

# AN ALGORITHM FOR RANDOM SIGNED 3-SAT WITH INTERVALS

KATHRIN BALLERSTEIN AND DIRK OLIVER THEIS

**ABSTRACT.** In signed  $k$ -SAT problems, one fixes a set  $M$  and a set  $\mathcal{S}$  of subsets of  $M$ , and is given a formula consisting of a disjunction of  $m$  clauses, each of which is a conjunction of  $k$  literals. Each literal is of the form “ $x \in S$ ”, where  $S \in \mathcal{S}$ , and  $x$  is one of  $n$  variables.

For Interval-SAT (iSAT),  $M$  is an ordered set and  $\mathcal{S}$  the set of intervals in  $M$ .

We propose an algorithm for 3-iSAT, and analyze it on uniformly random formulas. The algorithm follows the Unit Clause paradigm, enhanced by a (very limited) backtracking option. Using Wormald’s ODE method, we prove that, if  $m/n \leq 2.3$ , with high probability, our algorithm succeeds in finding an assignment of values to the variables satisfying the formula.

## 1. INTRODUCTION

Let  $M$  be a (usually finite) set,  $\mathcal{S}$  a set of subsets of  $M$ , and  $k$  a positive integer. For the *signed  $k$ -satisfiability problem*, or *signed  $k$ -SAT*, one is given as input a finite set of variables  $X$  and a formula in *signed conjunctive normal form (CNF)*. This means that there is a list of *clauses*, each of which is a conjunction ( $\wedge$ ) of (*signed*) *literals* of the form  $x \in S$  where  $x$  is a variable in  $X$  and the “sign”  $S$  is a set in  $\mathcal{S}$ . The question is then whether there exists a satisfying *interpretation*, i.e., an assignment of values to the variables such that each of the clauses is satisfied. This setting includes as a special case the classical satisfiability (SAT) problem. There, one chooses for  $M$  the 2-element set  $\{\text{TRUE}, \text{FALSE}\}$  and  $\mathcal{S} = \{\{\text{TRUE}\}, \{\text{FALSE}\}\}$ .

In case  $M$  is an ordered set (a chain) and the set  $\mathcal{S}$  is the set of all intervals in  $M$ , we speak of *Interval SAT*, or *iSAT*. In our contribution, we set  $M := [0, 1]$ , because this includes all iSAT settings with finite  $M$ . It is related to the study of random interval graphs [38, 32]. As is usually done in the context of SAT, we speak of  $k$ -iSAT, if each clause comprises  $k$  literals. Our notation and terminology on signed SAT follows [19].

Signed SAT problems originated in the area of so-called multi-valued logic [35], where variables can take a (usually finite) number of so-called *truth values*, not just TRUE or FALSE. Work on signed CNF formulas started in earnest with the work of Hähnle and Manyà and their coauthors. We refer the reader to the survey paper [10], and the references therein.

The motivation for studying signed formulas was to extend algorithmic techniques developed for deductive systems in multi-valued logic to better cover practical applications [29]. Indeed, on the one hand, a number of papers show how combinatorial problems can be solved using signed SAT algorithms [13, 14, 11, 26, 15]; on the other hand, a large number of heuristic and exact algorithms have been studied (see [5, 16] and the references therein), and a number of polynomially solvable subclasses of signed SAT have been identified [23, 10, 36, 9, 6, 5, 19]. While in the works of Manyà and his collaborators, order-theoretic properties of the ground

---

*Date:* Thu May 12 18:18:17 CEST 2011.

*2000 Mathematics Subject Classification.* Primary XXXXX.

set  $M$  are exploited to make conclusions on the complexity of signed SAT, Chepoi et al. [19] completely settle the complexity question in the general case by reverting to combinatorial properties of the set system  $\mathcal{S}$ . In particular, they prove that: signed  $k$ -SAT,  $k \geq 3$ , is polynomial, if, and only if,  $\bigcap_{S \in \mathcal{S}} S \neq \emptyset$  and NP-complete otherwise; signed 2-SAT is polynomial if, and only if,  $\mathcal{S}$  has the Helly-property (if no two sets in a subfamily are disjoint, then the subfamily has non-empty intersection), and NP-complete otherwise.

For the case when  $\mathcal{S}$  has the Helly-property, Chepoi et al. give a non-satisfiability certificate for signed 2-SAT in the spirit of Aspvall, Plass, and Tarjan’s famous result for classical 2-SAT [7].

Most applications and a great deal of the earlier complexity results [10] focus on *regular* signed SAT, where  $M$  is a poset, and the formulas may only involve sets of the form  $S = \{j \mid j \geq i\}$  or  $S = \{j \mid j \leq i\}$ . Regular iSAT (or just regular SAT) is regular signed SAT for posets  $M$  which are chains.

In particular, for regular iSAT, random formulas have been investigated from a heuristic point of view. Manyà et al. [37] study uniformly generated random regular 3-iSAT instances, and observe a phase transition similar to that observed in classical SAT (see [3] and the references therein): (i) the most computationally difficult instances tend to be found near the threshold, (ii) there is a sharp transition from satisfiable to unsatisfiable instances at the threshold and (iii) the value of the threshold increases as the number of truth values considered increases. Their results are confirmed and extended by further papers exploring uniformly random regular 3-iSAT instances [12, 10, 15].

Further, in [12, 15] a bound on the ratio  $m/n$  is given, beyond which a random formula is with high probability (whp) unsatisfiable. To our knowledge, however, ours is the first rigorous analysis of an algorithm for random signed SAT.

Our interest in the particular version of signed SAT arises from applications in computational systems biology, where iSAT yields a generalization of modeling with Boolean networks [33], where biological systems are represented by logical formulas with variables correspond to biological components like proteins. Reactions are modeled as logical conditions which have to hold simultaneously, and then transferred into CNF. The model is widely used by practitioners (see e.g. [22, 34, 30] and the references therein).

Often, though, this binary approach is not sufficient to model real life behavior or even accommodate all known data. Due to new measurement techniques, a typical situation is that an experiment yields several “activation levels” of a component. Thus, one wants to make statements of the form: If the quantity of component  $A$  reaches a certain threshold but does not exceed another, and component  $B$  occurs in sufficient quantity, then another component  $C$  is in a certain frame of activation levels. The collection of such rules accurately models the global behavior of the system. We refer to [8] for details of models and applications.

In this paper we present and analyze an algorithm which solves uniformly random 3-iSAT instances with high probability, provided that the ratio between the number  $m$  of clauses and the number  $n$  of variables is at most 2.3. Our algorithm is an adaption of the well-known Unit Clause algorithm from classical SAT [17, 2], where, in an inner loop, 1-clauses are treated if any exist, and in an outer loop, a variable is chosen freely and assigned some value. This Unit Clause approach is enhanced with a “backtracking” subroutine not very much unlike the

one studied in [25] for classical 3-SAT, where the currently best known algorithm for random classical  $k$ -SAT,  $k = 3$ , is given (but see [21] for general  $k$ ).

In the analysis, we use Wormald’s differential equations method [39]. ODE methods have been used for the analysis of algorithms for classical SAT with great success [17, 18, 25, 1, 4]. In our analysis, we combine the idea of Achlioptas and Sorkin [4] to consider as a time step an iteration of the outer loop, but we use Wormald’s theorem [40] where they use a Markov-chain based approach. The analysis of the inner loop requires to study the first busy period of a certain stable server system [1, 2], or, in our case, more accurately, the total population size in a type of branching process.

Moreover, it is important to point out that we prove that our algorithm succeeds with high probability, unlike [1, 4], where the aim<sup>1</sup> is only to prove success with positive probability (wpp). To obtain a whp result, the “backtracking” part is essential<sup>2</sup>, and its analysis has to be accommodated for, of course, in the analysis of the .

The value 2.3 arises from the numerical solution to an initial value problem (IVP). Extending the results for  $k$ -iSAT for  $k \geq 4$  is conceptually easy; we briefly discuss it in the conclusions.

The outline of the paper is as follows: In the next section, we present our algorithm for random 3-iSAT in detail. In Section 3, we prove some facts about uniformly at random chosen sub-intervals of  $[0, 1]$ . Section 4 we take a brief excursion to random 2-iSAT as our algorithm for 3-iSAT ultimately relies on solving a 2-iSAT instance. In Section 5, we compile the required facts about total population sizes of a kind of branching system, which are then applied in Section 6 to the study of the inner loop of our algorithm. Finally, in Section 7, we prove the whp-result for our algorithm. We raise some issues for future research in the final section. Several technical arguments have been moved into the appendix.

Throughout the paper, we hide absolute constants in the big- $O$ -notation. If the constant depends on other parameters, we make this clear by adding an index, e.g.,  $O_\varepsilon(\cdot)$ . As customary, we use the abbreviation iid for “independent and identically distributed” and uar for “uniformly at random”. Whp and wpp are to be understood for  $n \rightarrow \infty$ , with  $m = m(n)$  depending on  $n$ .

## 2. AN ALGORITHM FOR RANDOM 3-ISAT

In this section, we describe an algorithm which finds a satisfying interpretation if the number of clauses is  $m = cn$  with  $c \leq 2.3$ .

**2.1. The random model; exposure.** For our random model, we choose a formula uar from the set of all possible formulas on  $n$  variables with  $m$  3-clauses, each containing three distinct variables. As is customary in the context of random SAT, we use the language of “exposing” literals. Intuitively, the idea is that the information about each literal is written on a card which lies face down, until the information is exposed. Clearly, the unexposed part of the formula is uar conditioned on which literals have been exposed and which have not. We refer to the elegant description in Achlioptas paper [2].

<sup>1</sup>This was an artifact of the hunt for the precise value of the threshold in random  $k$ -SAT, for which, by Friedgut’s famous result [24], wpp statements suffice.

<sup>2</sup>Cf. for example, [25], where the range in which the algorithm succeeds increases dramatically, once a backtracking part is added.

**2.2. Brief description of the algorithm.** The basic framework of our Algorithm is the same as for most algorithms for classical  $k$ -SAT. A formerly unused variable is selected, and a value is assigned to it. Then, clauses containing the variable are updated: if the literal of the clause involving the variable is satisfied, the clause is deleted; otherwise the literal is deleted from the clause, leaving a shorter clause. The variable is removed from the set of *unused variables*, and declared a *used variable*. The algorithm fails if, and only if, it creates an empty clause.

However, to a certain extent, our algorithm is able to repair bad choices it has made. Thus, it occasionally only assigns *tentative* values to variables. As long as it is not certain that a variable keeps its tentative value, no deletions of clauses or literals from clauses are performed. Instead, we assign colors to the clauses, which code the number of satisfied, unsatisfied, and unexposed literals they contain. The meaning of the colors will be explained in Table 1 but at this point it suffices to know that red clauses correspond to unexposed 1-clauses, i.e., clauses with one unexposed literal and the variables in any other literal of the clause have tentative values which render the literals false.

As said before, the basic approach is that of the Unit-Clause algorithm. The *outer loop* of the algorithm will maintain the property that there is no 1-clause. In each iteration of the outer loop, a variable is selected from the set of unused variables. Such a variable selected in the outer loop is referred to as a *free variable*. The *inner loop* is initialized by assigning a tentative value to this free variable, and then repeats as long as there are red clauses. In each iteration of the inner loop, a red clause is selected and *served*: the variable contained in the clause (the *current variable* of the iteration) is tentatively set to some value in such a manner that the serviced red clause becomes true. We refer to the variables selected in the inner loop as *constrained variables*.

If, during a run of the inner loop, a situation is reached in which it is probable that an empty clause will be created, it backtracks. This happens when the following *fatality* is suffered: The current variable occurs in another red clause, other than the one serviced. If that happens, there is a  $1/3$  probability that the two intervals occurring in the two red clauses are disjoint [38], so that creating an empty clause is inevitable.

For this situation, the inner loop maintains a rooted tree  $G$  of decisions it has taken so far. The nodes of the tree correspond to variables to which tentative values have been assigned, with the root being the free variable with which the run of the inner loop was initialized. The edges correspond to clauses. The fatality entails that a circle is closed. If that happens the values of the variables along the paths from the root to serviced literal are changed. Then, all other tentative values are made permanent, and the inner loop is restarted with the new formula, but this time without a free variable in the initialization. We refer as *Phase I* to the run of the inner loop before a repair occurs (or if no repair occurs), and as *Phase II* to the run of the inner loop after a repair has been performed. In Phase II, no further repair is attempted. Instead, if fatalities occur, the inner loop takes a *Zen* approach, and just moves on (it “goes Zen”). In Phase I, if a fatality occurs, there’s the possibility that a repair is not possible. In this case, too, the inner loop “goes Zen”.

After all red clauses have been dealt with in either Phase I or Phase II, the tentative values are made permanent, and control is returned to the outer loop, which selects another free variable, and so on.

The outer loop terminates, if the number of 2-clauses plus the number of 3-clauses drops below a certain factor  $c'$  of the number of unused variables. Then, it deletes an arbitrary literal from every 3-clause and invokes the exact polynomial algorithm by Chepoi et al. [19] to decide whether the resulting 2-iSAT formula has a satisfying interpretation. We will prove in Section 4 that this is always the case if the ratio of the number of resulting 2-clauses over the number of unused variables is below  $\frac{3}{2}$ .

The complete algorithm is shown below as Algorithm 1 (the outer loop), Algorithm 2 (the inner loop), and Algorithm 3 (the repair procedure). Throughout the course of the algorithm, for  $i = 0, 1, 2, 3$ , we denote by  $Y_i(t)$  the number of  $i$ -clauses, and by  $X(t)$  the number of unused variables, respectively, at the beginning of iteration  $t$  of the outer loop. Moreover, for an interval  $I$ , we denote by

$$\bar{x}(I) := \operatorname{argmin}_{x \in I} |x - 1/2| \quad (1)$$

the point in  $I$  which is closest to  $1/2$ . We refer to the variable  $x_j$  which is selected in iteration  $j$  of the inner loop as the *current variable* of that iteration.

Below, we will prove the following fact.

**Lemma 1.** *A single run of Algorithm 2 (including a possible repair and consequent Phase II) produces an empty clause, only if it “goes Zen”.*

The performance of the algorithm on random 3-iSAT instances is analyzed in Sections 6 and 7. There, we will prove the following theorem.

**Theorem 2.** *Let  $c := 2.3$ , and suppose Algorithm 1 is applied to a uniformly random iSAT formula on  $n$  variables with  $m$  3-clauses. If  $m \leq cn$ , then, whp, Algorithm 1 creates no empty clause, i.e., it finds a satisfying interpretation.*

The value 2.3 is determined through the numerical solution of an initial value problem. It corresponds to the point in which the increase in red clauses in each iteration of the inner loop would become so large that the inner loop will not terminate.

---

**Algorithm 1** *UC w/ backtracking (outer loop)*

---

(o-1) Given: 3-CNF-formula; positive constant  $c'$ .

(o-2)  $t := 0$

(o-3) While  $Y_2(t) + Y_3(t) > c'X(t)$ :

    (o-3.1) Choose a variable  $x$  uar.

    (o-3.2) Invoke *Inner loop* (Phase I).

    (o-3.3)  $t := t + 1$

(o-4) In every 3-clause, remove one literal at random.

(o-5) Invoke Chepoi et al.’s algorithm (cf. Section 4) for the remaining 2-iSAT formula.

---

**2.3. Comparison to algorithms for classical SAT.** For classical SAT, if a variable  $x$  is set to a value, the probability that a random literal containing  $x$  evaluates to true is  $1/2$  — independently of the value. As will become apparent in the next section, this is far from true for random interval literals. There, the value  $1/2$  is the single, most likely value to be contained in a random interval (the probability is  $1/2$ ) and all other values are less likely. Hence, we will assign  $1/2$  to the variables as long as possible which is for all free variables.

**Algorithm 2** *Inner loop*

- 
- (i-1) Given:
- In **Phase I**: formula consisting of 2- and 3-clauses only; a (free) variable  $x_0$ .
  - In **Phase II**: formula consisting of 1-, 2- and 3-clauses.
- (i-2)  $j := 0$
- (i-3) Initialize: Expose the occurrences of  $x_0$  in all clauses.
- In **Phase I** only:
    - (i-3.1) Tentatively set  $x_0$  to  $1/2$ .
    - (i-3.2) Initialize the graph  $G := (\{x_0\}, \emptyset)$ .
  - In **Phase II** only:
    - (i-3.1) Color all 1-clauses red.
- (i-4) Expose the intervals associated with  $x_0$ . Color clauses containing  $x_0$  according to Tab. 1.
- (i-5)  $j := j + 1$
- (i-6) If there is no red clause, exit inner loop: Set all variables to their tentative values; remove satisfied clauses and remove violated literals from their clauses; return to outer loop.
- (i-7) Select a red clause  $C_j$  at random; let  $L_j$  be the unexposed literal in  $C_j$ ; expose current variable  $x_j$  of  $L_j$
- (i-8) Expose all occurrences of  $x_j$  in colored clauses.
- (i-9) If  $x_j$  is contained in a red clause other than  $C_j$ :
- In **Phase I** only:
    - (i-9.1) If there is a red, blue, or black 3-clause: go Zen!
    - (i-9.2) If the graph  $G$  contains a cycle, or  $x_j$  is in a blue clause: go Zen!
    - (i-9.3) If  $x_j$  occurs in three or more red clauses (including  $C_j$ ): go Zen!
    - (i-9.4) Otherwise: **Phase I** is completed. Let  $C'$  be the unique red clause different from  $C_j$  containing  $x_j$  in a literal  $L' = x_j \in J'$ . Repair the unique path between  $x_0$  and  $C_j$ ; then initiate **Phase II**.
  - In **Phase II** only: Go Zen!
- (i-10) Expose all occurrences of  $x_j$  in all uncolored clauses.
- (i-11) For every uncolored 2-clause  $x_j \in I \vee y \in J$  containing  $x_j$ , add to  $G$  the vertex  $y$  and the edge  $x_j \in I \vee y \in J$  between  $x_j$  and  $y$ .
- (i-12) Tentatively set  $x_j$  to  $\bar{x}(I_j)$ .
- (i-13) Update the colors of all clauses containing  $x_j$ .
- (i-14) Goto step (i-5).
- 

**Algorithm 3** *Repair path*

- 
- (r-1) Given: Set of colored 1-, 2- and 3-clauses; a literal  $L' = x_k \in J'$ ; a path of the form  $x_0, \quad x_0 \in J_0 \vee x_1 \in I_1, \quad x_1 \in J_1 \vee x_2 \in I_2, \quad \dots, \quad x_{k-1} \in J_{k-1} \vee x_k \in I_k$ ;
- (r-2) For  $j = 0, \dots, k-1$ :
- (r-2.1) Set  $x_j$  (permanently) to  $\bar{x}(J_j)$
- (r-3) Set  $x_k$  (permanently) to  $\bar{x}(J')$
- (r-4) Set all variables from Phase I, except those which have just been set in (r-2) and (r-3), to their tentative values; remove satisfied clauses and remove violated literals from their clauses.
-



Color	Meaning
Uncolored	All literals in the clause are unexposed.
Black	All literals are exposed.
Red	The clause has precisely one unexposed literal. The tentative values of any other variables in the clause make the corresponding literals false. In particular, unexposed 1-clauses are red.
Blue	The clause contains precisely one unexposed literal and at least one exposed literal which evaluates to true for the tentative value of its variable.
Pink	The clause is a 3-clause, precisely one of its literals is exposed, and this literal evaluates to false for the tentative value of its variable.
Turquoise	The clause is a 3-clause, precisely one of its literals is exposed, and this literal evaluates to true for the tentative value of its variable.

TABLE 1. Semantics of the colors of the clauses.

The rationale behind assigning the value  $1/2$  to free variables is two-fold. Firstly, it makes the analysis a lot more easy than if one tries to find a maximum cardinality subset of literals containing  $x$  all of whose intervals have pairwise non-empty intersection. Secondly, for large numbers of literals containing  $x$ , the maximum cardinality of a subset with pairwise intersecting intervals is asymptotically attained by taking all literals with intervals containing  $1/2$  (this is Theorem 4.7 of Scheinerman’s paper [38]).

The situation for constrained variables is similar, but a bit more complicated. For constrained variables, we are free only to choose the value for the variable within the interval  $I$  for the literal  $L = x \in I$  which we wish to satisfy. Unlike to classical SAT, where this does not change the probability that other random literals containing  $x$  are satisfied, depending on  $I$ , this probability may change considerably. Moreover, for two literals containing  $x$ , the two events of both being satisfied simultaneously with  $L$  are not independent.

However, an adaption of Scheinerman’s argument mentioned above shows that, asymptotically, the best choice is to take the point  $I$  which is closest to  $1/2$  as we do in our algorithm.

Concerning the backtracking part of the algorithm, we would like to point out the difference to the approach in [25]. If the (essentially identical) fatality is suffered, a very elegant remedy is to simply flip the values of all variables with tentative values: if the tentative value of a variable is TRUE, make it FALSE, and vice versa. Needless to say, for variable values in a larger set, there is no obvious choice for the new value of a variable. Thus, in our approach, we have to choose the variable values in a smart manner, with the single aim to undo the fatality. Namely, those variables that led to the fatality are assigned  $\bar{x}(I)$  as described in *Repair Path* (Algorithm 3).

#### 2.4. Proof of the Go-Zen lemma.

*Proof of Lemma 1.* Assume that Algorithm 2 does not go Zen.

The only place where a 0-clause can be generated without having gone Zen is in the final step 4 of the repair, Algorithm 3. Clearly, none of the clauses on the path will become empty.

Moreover, setting the final variable,  $x_k$ , cannot create an empty clause, because of the conditions in steps (i-9.1) and (i-9.2).

For a 3-clause to become empty, it is necessary that when the repair is invoked in Algorithm 2, all three of its literals have been exposed (possibly in the same iteration). In other words, it must have been red, blue, or black in step (i-9.1), a contradiction.

For a 2-clause to become empty, both literals must have been exposed, one of them possibly in the iteration where the repair occurs. Moreover, if it was blue, the value of the variable satisfying one of its literals must change during the repair. In other words, the following three scenarios are possible:

- (i) it was black before the repair was invoked
- (ii) it was red before the repair was invoked, but it contains  $x_j$
- (iii) it was blue before the repair was invoked, it is of the form  $x_i \in \mathcal{I}_i \vee x_j \in \mathcal{I}_j$  for some  $i < j$ , and  $x_i$  is one of the variables set in step (r-2) of Algorithm 3.

In case (i), if the black 2-clause becomes an empty clause, either it was red when its final literal was exposed, a contradiction, or it was blue, which means that at least one of its variables lies on the path which is repaired. If the whole clause lies on the path, we have already noted that it cannot become empty. If only one of its variables is on the path, then it must be an edge in the tree having one end vertex on the path and the other lying further away from the root than the path. The fact that it is black means that the variable which is not on the path was the current variable of some earlier iteration  $i < j$ . But then the corresponding literal was either the selected literal  $L_i$ , in which case it was satisfied by the tentative value of  $x_i$ , or the if-condition in step (i-9) for iteration  $i$  held, which is a contradiction (either a repair occurred, or the algorithm went Zen).

In case (ii), if the 2-clause is on the path, it does not become empty. If it is the unique other red clause  $C'$ , then it will be satisfied in the initialization of Phase II.

Case (iii), is not possible because of the condition in step (i-9.2) □

**2.5. Random formulas.** The following easy facts (see the discussion at the beginning of this section) underlies the analysis of the algorithm on random formulas.

**Lemma 3.** *If Algorithm 1 is invoked with a uar random 3-iSAT formula, then*

- (a) *at the beginning of each iteration of the outer loop, the current formula is distributed uar conditioned on the number of unused variables, 2-clauses, and 3-clauses;*
- (b) *at the beginning of each iteration of the inner loop, the current formula is distributed uar conditioned on the number of unused variables, 1-clauses, 2-clauses, 3-clauses, and the colors of the clauses.*
- (c) *at the beginning of Phase II in the inner loop, the current formula is not only conditioned on the number of unused variables, 1-clauses, 2-clauses, 3-clauses, the colors of the clauses, and the list  $L$  of clauses which are known not to contain  $x_0$  and the list of clauses in which an occurrence of  $x_0$  has been exposed.*

By Lemma 3, the history of the random process defined by the outer loop, that is, for each  $t$ , the state of the formula and all other information relevant to how the algorithm will proceed, available at the beginning of iteration  $t$ , is completely determined by

$$\mathcal{H}(t) := (X(t), Y_2(t), Y_3(t)); \tag{2}$$



in particular it is Markov.

### 3. COMPUTATIONS FOR RANDOM INTERVALS

In this section, we make some computations regarding intervals chosen uar from the subintervals of  $[0, 1]$ . We refer to [38, 32] for background.

We aim to study the event  $\bar{x}(I) \in J$ , with two random intervals  $I$  and  $J$  ( $\bar{x}$  is defined in (1)). We start with the following observation.

**Lemma 4** ([38]). *For  $x \in [0, 1]$  and for a random interval  $I$ , we have*

$$\mathbf{P}[x \in I] = 2x(1 - x).$$

*In particular, the probability that a random interval contains the point  $1/2$  is  $1/2$ .*

The cumulative distribution function of  $\bar{x}(I)$  can be written down.

**Lemma 5.** *For a random interval  $I$ , the random variable  $\bar{x}(I)$  has cumulative distribution function*

$$F(t) := \begin{cases} 0 & \text{if } t \leq 0 \\ t^2, & \text{if } 0 < t < 1/2 \\ 1 - (1 - t)^2, & \text{if } 1/2 \leq t < 1 \\ 1 & \text{if } t \geq 1. \end{cases} \quad (3)$$

*Proof.* Direct computation. □

Let  $X$  be a random variable with cumulative distribution function  $F$  as in (3), and define

$$P := 1 - 2X(1 - X). \quad (4)$$

Thus, by the previous two lemmas, the probability that, for two random intervals  $I$  and  $J$  we have  $\bar{x}(I) \in J$ , is

$$\mathbf{E}(\mathbf{P}[\bar{x}(I) \in J \mid P]) = \mathbf{E}(1 - P) = 1 - \mathbf{E} P.$$

The following computations are straightforward, see A.1.

**Lemma 6.**

$$(a) \quad \mathbf{E} P = 13/24$$

$$(b) \quad \mathbf{E} P^2 = 3/10$$
□

**Lemma 7.** *For two random intervals  $I, J$ , the following is true.*

$$\mathbf{P}[\bar{x}(I) \in J] = \frac{11}{24}.$$

*Proof.* Immediate from Lemmas 4, 5, and 6(a). □

#### 4. 2-iSAT

In this section, we take a brief glance at the situation for random 2-iSAT. The reason is that, ultimately, our 3-iSAT algorithm reduces the 3-iSAT formula to one with exactly two literals per clause, and then invokes the polynomial time algorithm by Chepoi et al. [19] to find a solution. We need to make sure that the resulting random 2-iSAT instance is satisfiable. We make no attempt to optimize the bound for  $m/n$  we obtain.

For this, we proceed along the same lines as [20], using Chepoi et al.'s Aspvall-Plass-Tarjan-type [7] certificate for the non-satisfiability of signed 2-SAT formulas for set systems satisfying the Helly-property. We describe the certificate now.

For a 2-iSAT formula  $F$ , define a digraph  $G_F$  which contains two vertices labeled  $xIt$  and  $xIf$ , respectively, for every literal  $x \in I$  occurring in  $F$ . For every clause  $x \in I \vee x' \in I'$  of  $F$ , the digraph  $G_F$  contains two arcs  $xIf \rightarrow x'I't$  and  $x'I'f \rightarrow xIt$ . We refer to these arcs as *clause arcs*. Moreover, for every two literals  $x \in I$  and  $x \in J$  occurring in  $F$ , if  $I \cap J = \emptyset$ , the digraph  $G_F$  contains the two arcs  $xIt \rightarrow xJf$  and  $xJt \rightarrow xIf$ . These arcs we call *disjointness arcs*.

For a literal  $x \in I$  occurring in  $F$ , we refer to the vertex  $xIt$  as a *positive* vertex, and to  $xIf$  as a *negative* vertex. Moreover, we say that these two vertices are *complements* of each other; in other words, the complement of the (positive) vertex  $xIt$  is the (negative) vertex  $xIf$  and vice versa. Note that arcs originating from negative vertices are clause arcs, while arcs originating from positive vertices are disjointness arcs.

Chepoi et al. relate the satisfiability of  $F$  to the strongly connected components (SCCs) of  $G_F$ .

**Proposition 8** (Aspvall-Plass-Tarjan-type certificate, [19]). *The formula  $F$  is satisfiable if, and only if, no SCC of  $G_F$  contains a pair of vertices which are complements of each other.*

*Remark 9.* A path in  $G_F$  of length  $\ell$  contains  $\lfloor \ell/2 \rfloor$  or  $\lceil \ell/2 \rceil$  disjointness arcs, and no two of them are incident.

Chepoi et al. also give an algorithm which determines, in polynomial time, whether a formula  $F$  is satisfiable, and if it is, produces a satisfying interpretation. We refer to their paper for details.

From Proposition 8, we obtain the following corollary.

**Corollary 10.** *If  $F$  is not satisfiable, then  $G_F$  contains a bicycle, i.e., a directed walk*

$$u_0 \rightarrow \cdots \rightarrow u_{\ell+1},$$

*with at least one clause-arc, and the following properties:*

- (a) *the literals in the vertices  $u_1, \dots, u_\ell$  are all distinct;*
- (b) *the literals in the vertices  $u_0$  and  $u_{\ell+1}$  occur among the literals in the other vertices;*
- (c) *the clauses in the arcs are all distinct.*

*Proof.* For a vertex  $v$ , we denote its complement by  $\bar{v}$ . By what we said about the different types of arcs, on every path from  $v$  to  $\bar{v}$ , there is at least one clause arc.

Choose an SCC and take a pair of complementing vertices  $v$  and  $\bar{v}$  in the SCC such that the distance from  $v$  to  $\bar{v}$  in  $G_F$  is minimal. Then, on the shortest path  $P$  from  $v$  to  $\bar{v}$ , no literal appears twice. Denote by  $L$  the literal defining  $v$  and  $\bar{v}$ .

Now take a shortest path  $Q$  in  $G_F$  from  $\bar{v}$  to  $v$ . If there is no literal other than  $L$  which appears twice on  $P \cup Q$ , then  $P \cup Q$  is a bicycle starting and ending in  $v$ . On the other hand, if there is a literal  $L'$  other than  $L$  which appears twice on  $P \cup Q$ , then the desired bicycle is constructed by taking the path  $P$  from  $v$  to  $\bar{v}$ , and then the path  $Q$  until the first vertex whose literal already occurred earlier.  $\square$

Suppose a 2-iSAT formula with  $n$  variables and  $m = cn$  clauses is drawn uniformly at random from the set of all such formulas (with the intervals all in  $[0, 1]$ ). We estimate the asymptotic probability that such a formula is satisfiable.

**Proposition 11.** *Let  $c' < 3/2$ . If  $m \leq c'n$  then, whp as  $n \rightarrow \infty$ , a randomly drawn 2-iSAT instance is satisfiable.*

The proof mimics that of Chvátal & Reed [20] for the classical 2-SAT very closely; we include it here just to point out where the number  $3/2$  comes in.

*Proof.* Given a fixed bicycle  $u_0 \rightarrow \dots \rightarrow u_{\ell+1}$  with  $r$  clause-arcs, the probability that it occurs in  $G_F$  is at most

$$\left( \frac{m}{\binom{n}{2}} \right)^r p^{r-1},$$

where  $p := 1/3$  is the probability that two independently chosen intervals are disjoint [38]. Hence, the expected number of bicycles with  $r$  clause-arcs occurring in  $G_F$  is at most

$$n^{r-1}(r-1)^2 \left( \frac{m}{\binom{n}{2}} \right)^r p^{r-1} \leq 2c' \frac{r^2}{n} (2c'p)^{r-1}$$

Thus, the expected total number of bicycles is at most

$$\frac{2c'}{n} \sum_{r=1}^{\infty} r^2 (2c'p)^{r-1}.$$

The sum is finite if, and only if,  $2c'p < 1$ , i.e.,  $c' < 3/2$ . Thus, in this case, the probability that a bicycle exists is  $O_{c'}(1)$ .  $\square$

Thus, for every  $c' < 3/2$ , whp, a satisfying interpretation can be found by Chepoi et al.'s algorithm [19].

## 5. TOTAL POPULATION SIZE OF OUR BRANCHING SYSTEM

As is done in classical SAT, the sub-routine eliminating the unit clauses can be viewed as a “discrete time” queue in which customers (i.e., unit clauses) arrive per time unit, the number depending on the customer currently serviced, and the single server, corresponding to one run of the inner loop of the algorithm, can process at least one customer per time unit. The number of iterations of the sub-routine then roughly corresponds to the length of the (first) busy period of the server.

Here, since, we are only interested in the length of the first busy period, the “queue” is really a branching system, for which for which we need to know the total number of individuals which are born before extinction. Compared to classical SAT, the interval-version poses several small challenges which we address in this section.

Let  $a$  be a non-negative integer, and  $B(j)$ ,  $j = 0, 1, 2, \dots$ , random variables taking values in the non-negative integers. We say the following sequence of random variables  $Q(j)$  a *discrete queue*:

$$\begin{aligned} Q(0) &= 0 \\ Q(1) &= a \\ Q(j+1) &= \begin{cases} a, & \text{if } Q(j) = 0 \\ Q(j) - 1 + B(j+1) & \text{if } Q(j) > 0 \end{cases} \end{aligned}$$

The number  $Q(j+1)$  is the number of individuals of the branching system after the  $j$ th individual has reproduced and died.

Denote by  $Z$  the length of the first busy period of the server, that is, the total population size of the branching process:

$$Z := \sup\{j \geq 0 \mid Q(i) > 0 \quad \forall i = 1, \dots, j\} = \inf\{j > 0 \mid Q(j) = 0\} - 1.$$

A straightforward adaption of the branching-process based textbook arguments for continuous-time M/G/1-queues gives the following (see A.2).

**Lemma 12.** *Suppose the  $B(j)$ ,  $j = 1, 2, \dots$ , are iid with mean  $\lambda_B$  and common probability generating function  $g_B$ . The probability generating function  $h$  of  $Z$  satisfies*

$$h\left(\frac{y}{g_B(y)}\right) = y^a \tag{5a}$$

for every  $y$  for which the power series  $g_B(y)$  converges and does not vanish. In particular, if  $\lambda_B < 1$ , we obtain

$$\mathbf{E} Z = \frac{a}{1 - \lambda_B}. \tag{5b}$$

Moreover, we have

$$\mathbf{P}[Z \geq \alpha] \leq \frac{g_B(y)^\alpha}{y^{\alpha-a}} \tag{5c}$$

for all  $\alpha > 0$  and  $y > 0$  with  $y \geq g_B(y)$ .  $\square$

*Remark 13.* Since we are only interested in the first busy period, we make the following modification to the definition of  $Q$ : If  $Q(j) = 0$  but  $j > 0$ , then we let  $Q(j+1) = 0$  (and not  $Q(j+1) = a$  as above). This makes some inequalities less cumbersome to write down.

**5.1. Bounding the tail probability for iid binomial  $B$ .** Let  $P$  be a random variable with values in  $[0, 1]$ . We say that a random variable  $B$  has binomial distribution with random parameter  $P$ , or  $\text{Bin}(m, P)$ , if

$$\mathbf{P}[B = k \mid P = p] = \binom{m}{k} p^k (1-p)^{m-k}.$$

In our setting  $n$  is a (large) integer, and  $m = m(n)$  is an integer depending on  $n$ . Define  $\lambda = \lambda(n) := \frac{m}{n}$ . Let  $P$  be as in (4), and suppose that  $B$  is  $\text{Bin}(m, 2^P/n)$ .

**Lemma 14.** *If  $\lambda(y-1) \leq 1/2$  we have*

$$g_B(y) \leq \exp\left(\frac{13}{12}\lambda(y-1) + \frac{6}{5}\lambda^2(y-1)^2\right)$$

*Proof.* We have  $e^t \leq 1 + t + t^2$  for all  $t \leq 1$ . For ease of notation, let  $\tau := \mathbf{E} P = 13/24$  and  $\tau_2 := \mathbf{E}(P^2) = 3/10$ , by Lemma 6. Since  $(y-1)\lambda 2P \leq 1$  with probability one, the following estimate holds:

$$\begin{aligned} g_B(y) &= \sum_{k=0}^m \mathbf{E} \left( \binom{m}{k} p^k \left(1 - \frac{2P}{n}\right)^{m-k} \right) = \mathbf{E} \left( \sum_{k=0}^m \binom{m}{k} p^k \left(1 - \frac{2P}{n}\right)^{m-k} \right) \\ &= \mathbf{E} \left( \left(1 + (y-1) \frac{2P}{n}\right)^m \right) \leq \mathbf{E} \left( e^{2(y-1)\lambda P} \right) \leq \mathbf{E} \left( 1 + 2(y-1)\lambda P + 4(y-1)^2 \lambda^2 P^2 \right) \\ &= 1 + 2\tau(y-1)\lambda + 4\tau_2(y-1)^2 \lambda^2 \leq e^{2\tau(y-1)\lambda + 4\tau_2(y-1)^2 \lambda^2} = \exp \left( \frac{13}{12} \lambda (y-1) + \frac{6}{5} \lambda^2 (y-1)^2 \right), \end{aligned}$$

as claimed.  $\square$

Now suppose that  $P(j)$ ,  $j = 1, 2, \dots$ , are iid random variables distributed as  $P$  defined in (4), and that  $B(j)$ ,  $j = 1, 2, \dots$ , are iid random variables distributed as  $\text{Bin}(m, 2P(j)/m)$ .

**Lemma 15.** *For every  $\varepsilon > 0$  there exist  $\delta > 0$  and  $C \geq 1$  such that, if  $1/2 \leq \frac{13}{12}\lambda \leq 1 - \varepsilon$ , the following is true.*

*For all  $\alpha \geq Ca$ , there exists a  $y$  with  $1 < g_B(y) < y \leq 2$  such that*

$$\frac{g_B(y)^\alpha}{y^{\alpha-a}} \leq e^{-\delta\alpha}. \quad (6)$$

*Proof.* For ease of notation, let  $u := y - 1$  and  $r := \frac{13}{12}\lambda$ , so that  $1/2 \leq r \leq 1 - \varepsilon$ . If  $0 < u < \frac{1-r}{r} \leq 1$ , by Lemma 14, we may estimate

$$g_B(y) \leq \exp \left( \frac{13}{12} \lambda u + \frac{6}{5} \lambda^2 u^2 \right),$$

and thus obtain

$$\mathbf{P}[Z \geq \alpha] \leq \exp \left( \alpha \left( \frac{13}{12} \lambda u + \frac{6}{5} \lambda^2 u^2 \right) - (\alpha - a) \log(u + 1) \right).$$

Using Lemma 6, we write the exponent as

$$\alpha r u + \frac{6 \cdot 12^2}{5 \cdot 13^2} \alpha r^2 u^2 - (\alpha - a) \log(u + 1). \quad (*)$$

In order to find a  $u$  minimizing (\*), we take the derivative and solve the resulting quadratic equation

$$\frac{12^3}{5 \cdot 13^2} r^2 u^2 + \left( r + \frac{12^3}{5 \cdot 13^2} r^2 \right) u - (1 - r) + a/\alpha = 0 \quad (**)$$

The value of  $u$  which works is the larger one of the two roots:

$$u_r := \frac{-\left(1 + \frac{12^3}{5 \cdot 13^2} r\right) + \sqrt{\left(1 - \frac{12^3}{5 \cdot 13^2} r\right)^2 + \frac{4 \cdot 12^3}{5 \cdot 13^2}}}{\frac{2 \cdot 12^3}{5 \cdot 13^2} r} - O(a/\alpha), \quad (***)$$

with an absolute constant in the  $O(\cdot)$  (see A.3 for the computation). The numerator is greater than zero if, and only if,  $4 \cdot \frac{12^3}{5 \cdot 13^2} r < \frac{4 \cdot 12^3}{5 \cdot 13^2}$ , which is equivalent to  $r < 1$ . Thus, there exists a  $C$  depending only on  $r$ , such that  $u_r > 0$  whenever  $\alpha \geq Ca$ . Moreover, by letting  $u = \frac{1-r}{r}$  in (\*\*), we see that  $u_r < \frac{1-r}{r} \leq 2$ , as required. Letting  $u = u_r$  in (\*), we obtain, for  $\alpha \geq Ca$ ,

$$(\delta_r(u_r) + O(1/C))\alpha, \quad (****)$$

with an absolute constant in the  $O(\cdot)$ , where

$$\delta_r(u) = ru + \frac{6 \cdot 12^2}{5 \cdot 13^2} r^2 u^2 - \log(u + 1)$$

(see A.3 for the computation). We have  $\delta_r(u_r) < 0$ , because  $\delta_r(0) = 0$  and since, by the choice of  $u_r$ , the derivative of  $\delta_r$  in the open interval  $[0, u_r[$  is negative. This also implies that  $y > g_B(y)$ . Let

$$\delta_* := \max\{\delta_r(u_r) \mid 1/2 \leq r \leq 1 - \varepsilon\} > 0,$$

Finally, increase  $C$ , if necessary, to take care of the dependence on  $O(1/C)$  in (\*\*\*) and (\*\*\*\*), and define  $\delta := -\delta_*/2$ . This completes the proof of the lemma.  $\square$

**Lemma 16.** *If  $\lambda \leq (1 - \varepsilon)\frac{12}{13}$ , then*

$$\mathbf{E} Z = \frac{a}{1 - \frac{13}{12}\lambda} \quad (7a)$$

and there exist  $\delta > 0$  and  $C \geq 1$  depending only on  $\varepsilon$ , such that for all  $\alpha \geq Ca$  we have the upper tail inequality

$$\mathbf{P}[Z \geq \alpha] \leq e^{-\delta\alpha}. \quad (7b)$$

*Proof.* Equation (7a) is directly from Lemma 12.

Lemmas 12 and 15 together imply the tail inequality in the case when  $\frac{13}{12}\lambda \geq 1/2$ . For smaller values of  $\lambda$ , we just note that increasing  $\lambda$  increases the length of the first busy period, so that the probability for  $\lambda := 6/13$  gives an upper bound for the probability for smaller values of  $\lambda$ .  $\square$

**5.2. Not-independent binomial.** The arrivals at the queue in the context of our algorithm are not completely independent. Here we deal with the small amount of dependence.

We now describe what kind of  $B(j)$  we allow. The setting is that  $n$  is a (large) integer, and that  $m = m(n) = \Theta(n)$ . Let  $r > 1$  and

$$z = z_r = z_r(n) := \frac{r}{\delta} \log n, \quad (8)$$

where  $\delta$  is as in Lemma 16. Suppose that  $M(j)$ ,  $N(j)$  are random variables satisfying

$$n - j \leq N(j) \leq n + j \quad \text{for all } j, \quad (9a)$$

$$0 \leq M(j) \leq m \quad \text{for all } j, \quad (9b)$$

with probability one, and

$$m^- \leq M(j) \leq m^+ \quad \text{for all } j = 1, \dots, z \quad (9c)$$

with probability at least  $1 - O(n^{-r})$ . Let the  $B(j)$  be distributed as  $\text{Bin}(M(j), \frac{P}{N(j)})$  for all  $j$ . More accurately, we assume that there is an iid family of  $P(j)$ ,  $j = 1, 2, 3, \dots$ , distributed as  $P$  above, and an independent family of random variables  $U(j, i)$ ,  $j = 1, 2, 3, \dots$ ,  $i = 1, 2, 3, \dots$  each having uniform distribution on  $[0, 1]$ , and that the joint distribution of the  $B(j)$  is the same as for the family of sums

$$\sum_{i=1}^{M(j)} \mathbf{I}\left[U(j, i) \leq \frac{P(j)}{N(j)}\right]. \quad (10)$$



The  $P(j)$  and  $U(j, i)$  are assumed to be jointly independent, but we make no assumptions about independence regarding the  $M(j)$  and  $N(j)$  among themselves or from the  $U(j, i)$  and  $P(j)$ . However, we do assume that  $a$ , the  $M(j)$ , and the  $N(j)$  are such that

$$a + \sum_{j=1}^{\infty} B(j) = O(n) \quad (11)$$

holds with probability one.

**Lemma 17.** *Let  $\lambda^{\pm} = \lambda^{\pm}(n) := \frac{m^{\pm}}{n - (\pm z)}$ , and suppose  $z \geq Ca$ . If*

$$\lambda^+ \leq (1 - \varepsilon) \frac{12}{13}, \quad (12)$$

*then with the  $\delta$  and  $C$  from Lemma 16, the following holds for large enough  $n$ :*

$$\frac{a}{1 - \frac{13}{12}\lambda^-} - O(n^{1-r}) \leq \mathbf{E} Z \leq \frac{a}{1 - \frac{13}{12}\lambda^+} + O(n^{1-r}); \quad (13a)$$

*and for all  $\alpha \geq Ca$*

$$\mathbf{P}[Z \geq \alpha] \leq e^{-\delta\alpha} + O(n^{-r}). \quad (13b)$$

The proof can be found in the appendix: A.4.

*Remark 18.* There is no danger in assuming  $\delta \leq 1$  and  $C \geq 1$ , and we will do that from this point on.

## 6. THE INNER LOOP

Here we analyze Algorithm 2. Conditioning on  $X(t)$ ,  $Y_2(t)$ , and  $Y_3(t)$ , we analyze the changes of the parameters  $X$ ,  $Y_2$ , and  $Y_3$  during the  $t + 1$ st run of the inner loop, and bound the probability that an empty clause is generated.

From now on,  $n$  and  $m$  denote the number of variables and clauses, respectively, in the initial random CNF formula, with  $m = cn$  for some constant  $c$ . We assume  $c \leq 10$ , to get rid of some of the letter  $c$  in the expressions below. For any  $\varepsilon > 0$ , we say that  $(x, y_2, y_3) \in \mathbb{R}^3$  is  $\varepsilon$ -good, if

$$\varepsilon n < x \quad \text{and} \quad \frac{y_2}{x} < (1 - \varepsilon) \frac{12}{13}, \quad (14)$$

and that  $\mathcal{H}(t)$  is  $\varepsilon$ -good if  $(X(t), Y_2(t), Y_3(t))$  is  $\varepsilon$ -good.

**6.1. Setup of the queues for Phases I and II.** We now define the queues for corresponding to the Phases I and II. We will suppress the dependency of the random processes on  $\mathcal{H}(t)$  in the notation.

We define the queues  $Q_I$  and  $Q_{II}$  for the Phases I and II, respectively, by modifying Algorithm 2 a little bit. We will then analyze (the original) Algorithm 2 with the help of the queues  $Q_I$  and  $Q_{II}$  defined via this modification. The changes we make are the following: replace step (i-7) by

(i-7') If there are unused variables left, choose one uar;

and step (i-8) by

(i-8') Expose all occurrences of the current variable  $x_j$  in clauses colored with a color different from red;

moreover, in the modification, we do not initiate a repair (since that would kill the queueing process).

Since, with these modifications, red clauses can contain used variables, it is possible to run out of variables before running out of clauses. It can be easily verified that this can only happen when all clauses are red. Hence, in this situation, the modified algorithm will just eat up the red clauses one per iteration.

In the Phase-I queue  $Q_I$ , the number of customers arriving in the first time interval,  $A_I$ , is the number of red clauses generated by setting the free variable  $x_0$  (tentatively) to  $1/2$ . Thus,  $A_I$  is distributed as  $\text{Bin}(Y_2(t), \frac{1}{X(t)})$ . For the iterations  $j = 1, 2, 3, \dots$ , we find that  $B_I(j+1)$  is the number of uncolored 2-clauses which become red, plus the number of pink 3-clauses which become red, when setting the current variable  $x_j$  (tentatively) to  $\bar{x}(\mathcal{I}_j)$ . Thus, if we denote by  $Y'_2(j)$  the number of uncolored 2-clauses plus the number of pink 3-clauses at the beginning of iteration  $j$ , then conditioned on  $Y'_2(j)$ , the distribution of  $B_I(j+1)$  is that of  $\text{Bin}(Y'_2(j), \frac{P(j+1)}{X(t)-j})$ , where as in the previous section, the  $P(j+1)$  are iid random variables distributed as  $P$  defined in (4). If we agree on the convention that a  $\text{Bin}(0, p/0)$ -variable is deterministically 0, this also holds when the queue runs out of variables.

In the Phase-II queue, the number of customers arriving in the first time interval,  $A_{II}$ , is the number of unit-clauses generated at the end of Phase I by setting the variables to their tentative values. The  $B_{II}(j)$  are defined analogous to the  $B_I(j)$ .

At this point, note that the condition (11), which is needed for Lemma 17, is satisfied for both queues.

**6.2. Bounds for the probabilities of some essential events.** Below, we repeatedly use the following simple Chernoff-type inequality (e.g. equation (2.11) in [31]): if  $U$  is a binomially distributed random variable with mean  $\mu$ , then

$$\mathbb{P}[U \geq \alpha] \leq e^{-\alpha} \quad \text{for } \alpha \geq 7\mu. \quad (15)$$

**Lemma 19.** *Let  $r > 1$ ,  $1 \leq z = z(n) = o(n)$  an integer,  $(x, y_2, y_3)$   $\varepsilon$ -good for some  $\varepsilon > 0$ , and  $m^- := \max(0, y_2 - rz \log n)$ ,  $m^+ := y_2 + rz \log n$ . For both phases I and II of the inner loop, the following is true. If, at the beginning of the phase at step (i-1), there are  $x$  variables,  $y_2$  2-clauses, and  $y_3$  3-clauses, then the probability that, while dealing with the first  $z$  variables in the phase, the number of 2-clauses leaves the interval  $[m^-, m^+]$ , is  $O(n^{-r})$ .*

*Proof.* For the upper bound  $m^+$ , the probability that the number of 2-clauses exceeds  $m^+$  is bounded from above by the probability that one in a sequence of  $z$  independent random variables with  $\text{Bin}(m, \frac{3}{\varepsilon n/2})$ -distributions is greater than  $r \log n$ . Here the factor  $1/2$  on the denominator takes care of the  $z = o(n)$  variables which are used. For  $n$  large enough, this probability is at most

$$zO\left(\binom{m}{r \log n} \left(\frac{6/\varepsilon}{n}\right)^{r \log n}\right) = zO(e^{-r \log n}) = O(n^{-r}).$$

For the lower bound  $m^-$ , the probability can be bounded by the same argument, noting that, if  $m^- = 0$ , the corresponding probability is 0.  $\square$

Let  $R$  denote the event that a repair is invoked during this run of Algorithm 2. Moreover, denote by  $Z_I$  and  $Z_{II}$  the length of the first busy period of the Phase I and Phase II queues, respectively. Note that they depend on  $A_I$  and  $A_{II}$ , respectively. Further let  $M_I$  and  $M_{II}$  be the total number of colored clauses which are generated during Phase I and Phase II, respectively; let  $H_I$  and  $H_{II}$  the event that, in some iteration, in steps (i-8), the current variable is found to be contained in a colored clause (other than the current clause  $C_j$ ); and by  $H_I^{\geq 2}$  the probability that in Phase I the current variable is found to be contained in at least two colored clauses (other than the current clause  $C_j$ ).

**Lemma 20.** *Suppose that  $\mathcal{H}(t)$  is  $2\varepsilon$ -good. With the  $\delta := \delta(\varepsilon)$  and  $C := C(\varepsilon)$  from Lemma 17, and  $r > 1$ , the following is true for all  $n$  large enough (depending on  $\varepsilon$ ).*

$$\mathbf{P}[A_I \geq r \log n \mid \mathcal{H}(t)] = O(n^{-r}) \quad (16a)$$

$$\mathbf{P}[Z_I \geq \frac{C}{\delta} r \log n \mid \mathcal{H}(t)] = O(n^{-r}) \quad (16b)$$

$$\mathbf{P}[M_I \geq \frac{500C}{\varepsilon\delta} r \log n \mid \mathcal{H}(t)] = O(n^{-r}) \quad (16c)$$

$$\mathbf{P}[H_I \mid \mathcal{H}(t)] = O_\varepsilon(\frac{\log^2 n}{n}) \quad (16d)$$

$$\mathbf{P}[H_I^{\geq 2} \mid \mathcal{H}(t)] = O_\varepsilon(\frac{\log^4 n}{n^2}) \quad (16e)$$

$$\mathbf{P}[R \mid \mathcal{H}(t)] = O_\varepsilon(\frac{\log^2 n}{n}) \quad (16f)$$

$$\mathbf{P}[A_{II} \geq \frac{500C}{\varepsilon\delta} (r+1) \log n \mid \mathcal{H}(t) \ \& \ R] = O(n^{-r}) \quad (16g)$$

$$\mathbf{P}[Z_{II} \geq \frac{500C^2}{\varepsilon\delta} (r+1) \log n \mid \mathcal{H}(t) \ \& \ R] = O(n^{-r}) \quad (16h)$$

$$\mathbf{P}[M_{II} \geq \frac{250000C^2}{\varepsilon^2\delta} (r+1) \log n \mid \mathcal{H}(t) \ \& \ R] = O(n^{-r}) \quad (16i)$$

$$\mathbf{P}[H_{II} \mid \mathcal{H}(t) \ \& \ R] = O_\varepsilon(\frac{\log^2 n}{n}) \quad (16j)$$

*Proof.* For (16a), if  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then the probability that  $A_I \geq \log n$  is bounded from above by the probability that a  $\text{Bin}(m, \frac{2}{2\varepsilon n})$ -variable is larger than  $r \log n$ , which is at most  $n^{-r}$ , for  $n$  large enough, by (15).

*Proof of (16b).* We use Lemma 17 together with Lemma 19 to bound the conditional probability that  $Z_I \geq \alpha$ . If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then the  $m^+$  from Lemma 19, with  $x := X(t)$ ,  $y_2 := Y_2(t)$ ,  $y_3 := Y_3(t)$ , and the  $z = z_r$  from (8), is such that (12) is satisfied if  $n$  is large enough depending on  $\varepsilon$ .

The requirement for the estimate in (13b) is that  $A_I \leq a_0 := \min(\alpha/C, z_r/C)$ . Thus, for the probabilities conditional on  $\mathcal{H}(t)$ , we have

$$\begin{aligned} \mathbf{P}[Z_I \geq \alpha] &= \mathbf{P}[Z_I \geq \alpha \mid A_I \leq a_0] \mathbf{P}[A_I \leq a_0] + \mathbf{P}[Z_I \geq \alpha \mid A_I > a_0] \mathbf{P}[A_I > a_0] \\ &\leq O(e^{-\delta\alpha}) + O(n^{-r}) + \mathbf{P}[A_I > a_0]. \end{aligned}$$

With  $\alpha := \frac{C}{\delta} r \log n$ , using (16a) and (15), the right-hand side is  $O(n^{-r})$ .

*Proof of (16c).* For every iteration, a clause is only colored if the current variable of the iteration is contained in the clause. Hence, the number of clauses colored in the first  $j$  iterations

is upper bounded by the sum of  $j$  independent  $\text{Bin}(m, \frac{3}{\varepsilon n})$ -variables. Hence, the probability that in the first  $j$  iterations, the number of colored variables exceeds  $j\alpha$  is at most  $e^{-\alpha}$  by (15), provided that  $\alpha \geq \frac{500}{\varepsilon}j \geq 7 \cdot \frac{3m}{\varepsilon n/2}j$ . Moreover, we have  $M_I \leq m$  with probability one. Thus, conditioning on  $\mathcal{H}(t)$  (and keeping in mind that  $\mathcal{H}(t)$  is required to be  $2\varepsilon$ -good), the probability that  $M_I$  is larger than  $\frac{500C}{\varepsilon\delta}r \log n$  is at most

$$O(e^{-r \frac{500C}{\varepsilon\delta} \log n}) + m \mathbf{P}[Z_I \geq r \frac{500C}{\varepsilon\delta} \log n \mid \mathcal{H}(t)] = O(n^{-r}) + O(mn^{-500r}) = O(n^{-r}).$$

*Proofs of (16d) and (16e).* In the first phase, in the  $j$ th iteration, the probability that the current variable  $x_j$  occurs in a colored clause (other than the current clause  $C_j$ ) is  $O(\frac{M_I}{X(t)-Z_I})$ , and the probability that the number of colored clauses containing  $x_j$  (other than the current one  $C_j$ ) is two or more is  $O((\frac{M_I}{X(t)-Z_I})^2)$ .

By (16b) and (16c), we can bound the probability that this happens in the first  $Z_I$  iterations by  $O_\varepsilon(\frac{\log^2 n}{n}) + O(n^{-r})$  and  $O_\varepsilon(\frac{\log^4 n}{n^2}) + O(n^{-r})$ , respectively, where the constant in the  $O_\varepsilon(\cdot)$  depends only on  $\varepsilon$ .

*Proof of (16f).* Clearly, the probability that a repair occurs is at most the probability that, in some iteration, the current variable  $x_j$  occurs in a colored clause (other than the current one  $C_j$ ). Thus, the inequality follow from (16d).

*Proof of (16g).* Since  $A_{II} \leq M_I$ , this inequality follows from (16c) and (16f), with  $r$  replaced by  $r+1$ , by conditioning on  $R$ :

$$\begin{aligned} \mathbf{P}[M_I \geq \frac{500C}{\varepsilon\delta}(r+1) \log n \mid \mathcal{H}(t) \ \& \ R] \\ &\leq \mathbf{P}[M_I \geq \frac{500C}{\varepsilon\delta}(r+1) \log n \mid \mathcal{H}(t)] / \mathbf{P}[R \mid \mathcal{H}(t)] \\ &= O(n^{-r-1} \frac{n}{\log^2 n}) = O(n^{-r}). \end{aligned}$$

*Proof of (16h).* We now apply Lemmas 17 and 19 to the Phase-II queue. Let  $r' := \frac{500C^2}{\varepsilon\delta}(r+1)$ . If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then the  $m^+$  from Lemma 19, with  $x := X(t)$ ,  $y_2 := Y_2(t)$ ,  $y_3 := Y_3(t)$ , and the  $z = z_{r'}$  from (8), is such that (12) is satisfied if  $n$  is large enough depending on  $\varepsilon$ .

Again, the requirement for the estimate in (13b) is that  $A_{II} \leq a'_0 := \min(\alpha/C, z_{r'}/C)$ . Thus, for the probabilities conditional on  $\mathcal{H}(t) \ \& \ R$ , we have

$$\begin{aligned} \mathbf{P}[Z_{II} \geq \alpha] \\ &= \mathbf{P}[Z_{II} \geq \alpha \mid A_{II} \leq a_0] \mathbf{P}[A_{II} \leq a'_0] + \mathbf{P}[Z_{II} \geq \alpha \mid A_{II} > a_0] \mathbf{P}[A_{II} > a'_0] \\ &\leq O(e^{-\delta\alpha}) + O(n^{-r'}) + \mathbf{P}[A_{II} > a'_0] \end{aligned}$$

With  $\alpha := \frac{500C^2}{\varepsilon\delta}(r+1) \log n$ , we have  $a'_0 = \frac{500C}{\varepsilon\delta}(r+1) \log n$ , so that, by (16g), the probability that  $A_{II} > a'_0$  is  $O(n^{-r})$ . In total, we obtain an upper bound of  $O(n^{-r})$  for the probability that  $Z_{II} \geq \frac{500C^2}{\varepsilon\delta}(r+1) \log n$ .

*Proof of (16i).* For every iteration, a clause is only colored if the current variable of the iteration is contained in the clause. Hence, the number of clauses colored in the first  $j$  iterations is upper bounded by the sum of  $j$  independent  $\text{Bin}(m, \frac{3}{\varepsilon n/2})$ -variables. (The factor of  $1/2$  in the denominator is to take care of the fact that the number of variables, while starting with at least

$\varepsilon n$ , might drop below  $\varepsilon n$  during the run of Phase I or Phase II.) Hence, the probability that in the first  $j$  iterations, the number of colored variables exceeds  $j\alpha$  is at most  $e^{-\alpha}$  by (15), provided that  $\alpha \geq \frac{500}{\varepsilon}j \geq 7 \cdot \frac{3m}{\varepsilon n/2}j$ . Moreover, we have  $M_{II} \leq m$  with probability one. Thus, conditioning on  $\mathcal{H}(t)$  &  $R$  (and keeping in mind that  $\mathcal{H}(t)$  is  $2\varepsilon$ -good), the probability that  $M_{II}$  is larger than  $\frac{500^2 C^2}{\varepsilon^2 \delta}(r+1) \log n$  is at most

$$\begin{aligned} O(e^{-\frac{500^2 C^2}{\varepsilon^2 \delta}(r+1) \log n}) + m \mathbf{P}[Z_{II} \geq \frac{500^2 C^2}{\varepsilon^2 \delta}(r+1) \log n \mid \mathcal{H}(t)] \\ = O(n^{-r}) + O(mn^{-500r}) = O(n^{-r}). \end{aligned}$$

*Proof of (16j).* In the second phase, in the  $j$ th iteration, the probability that the current variable  $x_j$  occurs in a colored clause (other than the current one  $C_j$ ) is  $O(\frac{M_I}{X(t)-Z_I})$ . By (16h) and (16i), we can bound the probability that this happens in the first  $Z_{II}$  iterations by  $O_\varepsilon(\frac{\log^2 n}{n}) + O(n^{-r})$ , where the constant in the  $O_\varepsilon(\cdot)$  depends only on  $\varepsilon$ .  $\square$

**6.3. Changes of the parameters  $X(t)$ ,  $Y_2(t)$ , and  $Y_3(t)$ .** We now move to study the differences between successive values of these parameters, and we start with  $X(t+1) - X(t)$ . Denote by  $F_I$  and  $F_{II}$  the number of iterations of the inner loop in the first and second phase, respectively. Clearly,  $X(t) - X(t+1) = 1 + F_I + F_{II}$ , where the leading 1 accounts for the free variable  $x_0$ . Moreover, we have  $F_I \leq Z_I$  and  $F_{II} \leq Z_{II}$ , and the inequality can be strict for two reasons: in Phase I, a repair can occur, thus terminating the phase before  $Q_I$  drops to zero; in both phases a red clause can vanish (i.e. become black) in (i-9). However, note that

$$\begin{aligned} F_I &= Z_I && \text{with probability } 1 - O_\varepsilon(\frac{\log^2 n}{n}), \text{ and} \\ F_I \mathbf{I}[\overline{R}] &\geq Z_I \mathbf{I}[\overline{R}] - 1 && \text{with probability } 1 - O_\varepsilon(\frac{\log^4 n}{n^2}) \end{aligned} \quad (17)$$

by (16d), (16f) and (16e).

Let us abbreviate

$$\Delta X := -1 - \frac{\frac{Y_2(t)}{X(t)}}{1 - \frac{13Y_2(t)}{12X(t)}} = -1 - \frac{12Y_2(t)}{12X(t) - 13Y_2(t)} = -\frac{12X(t) - Y_2(t)}{12X(t) - 13Y_2(t)}.$$

**Lemma 21.** *If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good and  $n$  large enough depending on  $\varepsilon$ , then*

$$\left| 1 - \Delta X - \mathbf{E}(Z_I \mid \mathcal{H}(t)) \right| = O_\varepsilon(\frac{\log^2 n}{n}) \quad (18a)$$

$$\left| \Delta X - \mathbf{E}(X(t+1) - X(t) \mid \mathcal{H}(t)) \right| = O_\varepsilon(\frac{\log^4 n}{n}) \quad (18b)$$

and

$$\mathbf{P} \left[ |X(t+1) - X(t)| \geq \log^2 n \mid \mathcal{H}(t) \right] = O(n^{-10}) \quad (18c)$$

*Proof.* By what we have said above on the relationship between  $F_I$ ,  $F_{II}$  and  $X(t+1) - X(t)$ , we have  $F_I = Z_I \mathbf{I}[\overline{R}] - E_I$  and  $F_{II} = Z_{II} - E_{II}$ , where  $E_I$  and  $E_{II}$  are error terms accounting for red clauses vanishing. We have  $\mathbf{E}(E_I \mid \mathcal{H}(t)), \mathbf{E}(E_{II} \mid \mathcal{H}(t)) = O_\varepsilon(\frac{\log^4 n}{n})$  by (16d) and (16e) (noting that  $E_I, E_{II} \leq m$ ).

We compute the the mean of  $Z_I$  using Lemma 17 with the  $m^\pm$  from Lemma 19 with  $z := \frac{r}{\delta} \log n$  as in (8). Thus, letting  $v := rz \log n$  (the bound from Lemma 19), for each  $a$ , conditional on  $A_I$  and  $\mathcal{H}(t)$ , we have

$$\frac{A_I}{1 - \frac{13Y_2(t)-v}{12X(t)+z}} \leq \mathbf{E}(Z_I \mid A_I \ \& \ \mathcal{H}(t)) \leq \frac{A_I}{1 - \frac{13Y_2(t)+v}{12X(t)-z}},$$

so that

$$\mathbf{E}(Z_I \mid A_I \ \& \ \mathcal{H}(t)) = \frac{A_I}{1 - \frac{13Y_2(t)}{12X(t)}} + O_\varepsilon\left(\frac{A_I \log^2 n}{n}\right),$$

provided that  $A_I \leq z/C$ , which holds with probability at least  $1 - O(n^{-2})$  by (16a) by increasing, if necessary,  $r$  beyond  $2\delta C$ . Since  $Z_I = O(n)$  with probability one, we obtain

$$\begin{aligned} \mathbf{E}(Z_I \mid \mathcal{H}(t)) &= \mathbf{E}\left(\frac{A_I}{1 - \frac{13Y_2(t)}{12X(t)}} + O_\varepsilon\left(\frac{A_I \log^2 n}{n}\right) \mid \mathcal{H}(t)\right) \\ &= \frac{\mathbf{E}(A_I \mid \mathcal{H}(t))}{1 - \frac{13Y_2(t)}{12X(t)}} + O_\varepsilon\left(\frac{\log^2 n}{n}\right) = \frac{\frac{Y_2(t)}{X(t)}}{1 - \frac{13Y_2(t)}{12X(t)}} + O_\varepsilon\left(\frac{\log^2 n}{n}\right), \end{aligned}$$

which proves (18a). For  $F_I$ , we obtain

$$\begin{aligned} \mathbf{E}(F_I \mid \mathcal{H}(t)) &= \mathbf{E}(Z_I \mid \mathcal{H}(t)) - \mathbf{E}(Z_I \mathbf{I}(R) \mid \mathcal{H}(t)) - \mathbf{E}(E_I \mid \mathcal{H}(t)) \\ &= \Delta X + O_\varepsilon\left(\frac{\log^2 n}{n}\right) - O_\varepsilon(\log n) \mathbf{P}(R \mid \mathcal{H}(t)) - mO(n^{-r}) - O_\varepsilon\left(\frac{\log^4 n}{n}\right) \\ &= \Delta X + O_\varepsilon\left(\frac{\log^4 n}{n}\right) \end{aligned}$$

and

$$\mathbf{E}(F_{II} \mid \mathcal{H}(t) \ \& \ R) \leq \mathbf{E}(Z_{II}) = O_\varepsilon(\log n) + O(n^{-r})m,$$

from which (18b) follows.

Since  $X(t) - X(t+1) \leq 1 + F_I + F_{II}$ , the tail inequality (18b) follows immediately from (16b) and (16h).  $\square$

**Lemma 22.** *If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then*

$$\left| -\Delta X \frac{3Y_3(t)}{X(t)} - \mathbf{E}(Y_3(t+1) - Y_3(t) \mid \mathcal{H}(t)) \right| = O_\varepsilon\left(\frac{\log^4 n}{n}\right) \quad (19a)$$

and

$$\mathbf{P}\left[|Y_3(t+1) - Y_3(t)| \geq \log^2 n \mid \mathcal{H}(t)\right] = O(n^{-10}) \quad (19b)$$

*Proof.* Let us denote by  $X'(j)$  the number of unused variables after  $j$  iterations of the inner loop, i.e., before  $\mathbf{x}_j$  is used, for  $j = 0, 1, 2, \dots$ . In every iteration of the inner loop, regardless of whether in Phase I or Phase II, for every uncolored 3-clause  $C$ , there is a  $\frac{3}{X'(j)}$  probability that the current variable  $\mathbf{x}_j$  is found to be contained in  $C$  in step (i-6.10), or (i-3.3), respectively, for the zeroth iteration in Phase I. If that is the case, the 3-clause is colored, and when the inner loop terminates, the clause will no longer be a 3-clause.



If we suppose that, at the beginning of iteration  $j = 0, 1, 2, \dots$ , before the current variable  $x_j$  is treated, there are  $Y'_3(j)$  uncolored 3-clauses and  $X'(j)$  unused variables, then the number of 3-clauses which are hit  $x_j$  is distributed as  $\text{Bin}(Y'_3(j), 3/X'(j))$ . (We have  $X'(j) = X(t) - j$  in Phase I, but in Phase II the value of course depends on how Phase I went.)

For (19b), we can just use the fact that the number of 3-clauses which are colored is bounded from above by  $M_I + M_{II}$ , the total number of colored clauses. Thus, by (16c) and (16i), this number is at most  $\log^2 n$  with probability  $1 - O(n^{-10})$  for  $n$  large enough depending on  $\varepsilon$ .

For the conditional expectation estimate (19a), we compute, conditional on  $\mathcal{H}(t)$ ,

$$\mathbf{E}(Y_3(t+1) - Y_3(t)) = \mathbf{E}((Y_3(t+1) - Y_3(t)) \mathbf{I}[R]) + \mathbf{E}((Y_3(t+1) - Y_3(t)) \mathbf{I}[\bar{R}]).$$

For the left summand, we have

$$\begin{aligned} \mathbf{E}((Y_3(t+1) - Y_3(t)) \mathbf{I}[R]) &\leq \mathbf{E}(\log^2 n \mathbf{I}[R \text{ \& } Y_3(t+1) - Y_3(t) \leq \log^2 n]) \\ &\quad + \mathbf{E}(m \mathbf{I}[R \text{ \& } Y_3(t+1) - Y_3(t) \geq \log^2 n]) \\ &\leq \log^2 n \mathbf{P}[R] + m \mathbf{P}[Y_3(t+1) - Y_3(t) \geq \log^2 n] = \log^2 n O_\varepsilon\left(\frac{\log^2 n}{n}\right) + O(n^{-9}) \\ &= O_\varepsilon\left(\frac{\log^4 n}{n}\right), \end{aligned}$$

by (16f) and (19b).

For the right summand, we have

$$\mathbf{E}((Y_3(t+1) - Y_3(t)) \mathbf{I}[\bar{R}]) = \mathbf{E}\left(\sum_{j=1}^{Z_I+1} G(j) \mathbf{I}[\bar{R}]\right) + O_\varepsilon\left(\frac{\log^2 n}{n}\right),$$

where, conditioned on  $Y'_3(j)$  as defined above, the  $G(j+1)$  are distributed as  $\text{Bin}(Y'_3(j), \frac{3}{X(t)-j})$ , and the  $O(\cdot)$  accounts for the possibility that  $F_I < Z_I$ , cf. (17). Using (15) and a similar argument as above, we see that

$$\mathbf{E}\left(\sum_{j=1}^{Z_I+1} G(j) \mathbf{I}[\bar{R}]\right) = \mathbf{E}\left(\sum_{j=1}^{Z_I+1} G(j)\right) + O_\varepsilon\left(\frac{\log^4 n}{n}\right).$$

Computing the expectation of the sum can be done in the same way as for classical SAT (e.g. in [1, 4, 2]). Indeed, using the optional stopping theorem ( $Z_I + 1$  is a stopping time for the history of the queue together with all random processes involved; cf. the proof of the next lemma for the details, where the situation is essentially the same, only a bit more complicated), we find that

$$\mathbf{E}\left(\sum_{j=1}^{Z_I+1} G(j)\right) = \mathbf{E}\left(\sum_{j=0}^{Z_I} \frac{3U(j)}{X(t)-j}\right),$$

where we agree that  $0/0 = 0$ . By (19b),  $Y_3(t) - \log^2 n \leq Y'_3(j) \leq Y_3(t)$  with probability  $1 - O(n^{-9})$ , and by (16b) we have  $Z_I \leq \log^2 n$ , implying  $X(t) - j \geq \frac{1}{2}X(t)$ , with probability

$1 - O(n^{-10})$ . Thus, we conclude

$$\begin{aligned}
& \mathbf{E} \left( \sum_{j=0}^{Z_I} \frac{3Y'_3(j)}{X(t) - j} \right) \\
&= \mathbf{E} \left( \mathbf{I}[Y_3(t) - \log^2 n \leq Y'_3(j) \ \& \ X(t) - j \geq \tfrac{1}{2}X(t)] \cdot \sum_{j=0}^{Z_I} \left( \frac{3Y_3(t)}{X(t)} + O\left(\frac{X(t) \log^2 n}{X(t)^2}\right) \right) \right) \\
&\quad + O(n^{-7}) \\
&= (1 + \mathbf{E} Z_I) \left( \frac{3Y_3(t)}{X(t)} + O\left(\frac{\log^2 n}{n}\right) + O(n^{-7}) \right) = -\Delta X \frac{3Y_3(t)}{X(t)} + O\left(\frac{\log^2 n}{n}\right),
\end{aligned}$$

by (18a). This concludes the proof of (19a).  $\square$

**Lemma 23.** *If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then*

$$\left| \frac{3Y_3(t)}{2X(t)} + (\Delta X - 1) \frac{13Y_3(t)}{8X(t)} - \Delta X \frac{2Y_2(t)}{X(t)} - \mathbf{E}(Y_2(t+1) - Y_2(t) \mid \mathcal{H}(t)) \right| = O_\varepsilon\left(\frac{\log^4 n}{n}\right) \quad (20a)$$

and

$$\mathbf{P} \left[ |Y_2(t+1) - Y_2(t)| \geq \log^2 n \mid \mathcal{H}(t) \right] = O(n^{-10}) \quad (20b)$$

*Proof.* The tail inequality is obtained by referring to (16c) and (16i) again, since very clause which changes its length has been colored before that can happen.

Let us denote by  $X'(j)$  the number of unused variables after  $j$  iterations of the inner loop, i.e., before  $\mathbf{x}_j$  is selected. In every iteration of the inner loop, regardless of whether in Phase I or Phase II, for every uncolored 2-clause  $C$ , there is a  $\frac{2}{X'(j)}$  probability that the current variable  $\mathbf{x}_j$  is found to be contained in  $C$  in step (i-6.10), or (i-3.3), respectively, for the zeroth iteration in Phase I. If that is the case, the 2-clause is colored, and when the inner loop terminates, the clause will no longer be a 2-clause. The same is true for 3-clauses which have become red in some previous iteration. Denote the total number of 2-clauses and pink 3-clauses which are hit by the current variable in some iteration over the whole run of Algorithm 2 by  $L_{2\times}$ .

The analysis of the expectation and tail of  $L_{2\times}$  is almost identical to the analysis done in the previous lemma for the 3-clauses. Here, too, we have to condition on the number of uncolored 2-clauses and pink 3-clauses not changing too much. The difference is the need to control the number of pink 3-clauses and, after a repair, the number of 3-clauses becoming 2-clauses. The latter two numbers are bounded from above by  $Y_3(t+1) - Y_3(t)$ , which is at most  $\log^2 n$  with probability  $1 - O_\varepsilon(n^{-10})$ . Thus, for  $L_{2\times}$ , we just note that its expectation accounts for the summand  $-\Delta X \frac{2Y_2(t)}{X(t)}$  in (20a).

Now let us denote the number of 3-clauses which become 2-clauses during the two phases of the inner loop by  $L_{3 \rightarrow 2}$ , and let us also focus on the case when no repair occurs.

In this case  $L_{3 \rightarrow 2}$  behaves similarly to  $Y_3(t+1) - Y_3(t)$ , with two differences: The probabilities that a 3-clause is colored pink is different; and the zeroth iteration differs from the

others. Let us first consider the zeroth iteration. The probability that the tentative value  $1/2$  of  $x_0$  makes a 3-clause pink is  $1/2$  by Lemma 4. Thus, if there is no repair, this contribution is distributed as  $\text{Bin}(Y_3(t), \frac{1}{2} \cdot \frac{3}{X(t)})$ .

For the other iterations,  $j = 1, 2, 3, \dots$ , if an uncolored 3-clause  $C$  contains the current variable  $x_j$ , the probability that  $C$  becomes pink in (i-13) depends on the current interval  $\mathbb{I}_j$ , and is distributed as  $P$  defined in (4). Indeed, if we denote the number of uncolored 3-clauses in iteration  $j$  by  $Y'_3(j)$  again, then, conditioned on  $Y'_3(j)$  and  $X'(j)$ , the number  $G(j+1)$  of uncolored 3-clauses which become pink in iteration  $j$  is distributed as  $\text{Bin}(Y'_3(j), \frac{3P(j+1)}{X'(j)})$ , i.e., binomial with random parameter  $P(j+1)$ . The  $P(j)$  are the iid random variables distributed as  $P$  in (4) defined by  $\bar{x}(\mathbb{I}_j)$ , in other words  $P(j+1) = 1 - 2\bar{x}(\mathbb{I}_j)(1 - \bar{x}(\mathbb{I}_j))$ .

Let  $G(1)$  be distributed as  $\text{Bin}(Y_3(t), \frac{3}{2X(t)})$ , define  $D(j+1) := G(j+1) - \frac{13Y'_3(j)}{8X'(j)}$ , where we agree that  $0/0 = 0$ , and denote by  $\mathcal{F}(j)$  the history of the process up to iteration  $j$ , i.e., before the variable  $x_j$  is treated. Then  $\sum_{j=1}^{\ell} D(j)$ ,  $\ell = 1, 2, 3, \dots$ , is a martingale with respect to  $\mathcal{F}(j)$ ,  $j = 0, 1, 2, \dots$ , and  $Z_I + 1$  is a stopping time, because deciding whether  $Z_I + 1 \leq \ell$  amounts to checking whether  $Q_I(\ell) = 0$ .

To estimate the expectation of the contribution of these, we use the optional stopping theorem again; note that the stopping time is finite with probability one, because  $Z_I \leq m$ . We conclude that  $\mathbf{E} \left( \sum_{j=1}^{Z_I+1} D(j) \right) = 0$ , which means

$$\mathbf{E} \left( \sum_{j=1}^{Z_I+1} G(j) \right) = \mathbf{E} \left( \sum_{j=0}^{Z_I} \frac{13Y'_3(j)}{8X'(j)} \right).$$

Arguing as we have done a number of times in regard of the possible deviations of  $Y'(j)$  from  $Y(t)$ , we see that the right hand side equals

$$(\mathbf{E} Z_I + 1) \frac{13Y_3(t)}{8X(t)} + O_{\varepsilon} \left( \frac{\log^4 n}{n} \right).$$

Getting rid of the conditioning on the event that no repair occurs is done in the same way as in the previous lemma, and we leave the details to the reader.  $\square$

**6.4. Failure probability.** We now bound the probability that an empty clause is generated by a run of the inner loop, including, possibly, the repair and following second phase.

**Lemma 24.** *If  $\mathcal{H}(t)$  is  $2\varepsilon$ -good, then the probability that Algorithm 2 produces an empty clause, is  $o(1/n)$ .*

*Proof.* We use Lemma 1. Let us first deal with Phase II. The probability that the algorithm goes Zen in Phase II is  $O_{\varepsilon}(\frac{\log^2 n}{n})$  by (16j), conditioned on a repair occurring, so that by the law of total probability, the probability that the algorithm goes Zen in Phase II is at most  $O_{\varepsilon}(\frac{\log^4 n}{n^2})$ , by (16f).

For Phase I, we need to go through the possible reasons for the algorithm to go Zen. First of all, by (16e), the probability that the current variable  $x_j$  is contained in a colored (red or not) clause other than the current one  $C_j$  is  $O_{\varepsilon}(\frac{\log^4 n}{n^2})$ , which takes care of step (i-6.3).

The probability that a fixed clause contains the current variable of a fixed iteration depends only on the number of variables and the number of unexposed atoms in the clause, and so it

can always be bounded by  $\frac{3}{\varepsilon n}$ . In order for a 3-clause to become red or blue (or even black), it must contain the current variable of (at least) two iterations. The probability of this happening is  $O_\varepsilon(\frac{\log n}{n^2})$ , where we have used (16b). This gives the case of step (i-9.1).

Similarly, for step (i-9.2), a 2-clause must have been hit twice by the current variable of an iteration, the probability of which is again bounded by  $O_\varepsilon(\frac{\log n}{n^2})$ .

In total, the failure probability can be bounded by  $O(\frac{\text{polylog } n}{n^2})$   $\square$

## 7. THE OUTER LOOP

At the heart of analysis of the outer loop is the well-known theorem of Wormald's which, in certain situations, allows to estimate parameters of random processes by solutions to differential equations. Here is the first goal of our analysis.

**Lemma 25.** *For every  $c \in ]0, 3]$ , the initial value problem*

$$\frac{dy}{dx} = \frac{-18cx^4 + 2y(12x - y)}{x(12x - y)} \quad (21a)$$

$$y(1) = 0 \quad (21b)$$

*has a unique solution  $y$  defined on the interval  $]0, 1]$ .*

See Fig 1 for a rough sketch of the direction field (21a) with  $c = 2$ , and a solution to the IVP. Since, ultimately, we will solve the IVP (21) numerically for the right value of  $c$  anyway, strictly speaking, this lemma is not needed to complete our argument. However, we would like to reduce our reliance on numerical computations as much as possible.

*Proof of Lemma 25.* To use the known theorems on IVPs, note that the right hand side of (21a), seen as a function of  $(x, y)$ , is continuously differentiable on  $\{(x, y) \mid x > 0, y < 12x\}$ .

We make the following claims:

- (a) *For  $4/5 \leq x \leq 1$ , the solution to the IVP never crosses the line  $y = 5(1 - x)$ ;*
- (b) *for  $0 < x \leq 4/5$ , the solution to the IVP never crosses the line  $y = 6x$ .*

Thus, the solution to the IVP does not approach the  $y = 12x$ , which implies that the solution extends to the whole interval  $]0, 1]$ .

Let  $g(x, y) := \frac{-18cx^4 + 2y(12x - y)}{x(12x - y)}$ , the right hand side of the ODE (21a). To prove claim (b), it suffices to show that, with  $y(x) := 6x$ , whenever  $0 < x \leq 4/5$ , we have  $\frac{dy}{dx} < g(x, y(x))$ . The computation is easy but tedious and can be found in the appendix, see A.5. Similarly, for claim (a), with  $y(x) := 5(1 - x)$ , for every  $4/5 \leq x \leq 1$ , we have  $\frac{dy}{dx} < g(x, y(x))$ . The computation is in the appendix, too.  $\square$

**Lemma 26.** *Let  $c \leq 3$  and  $y$  a solution to (21), and let  $x_0$  be the infimum over all  $x \geq 3\varepsilon$  for which*

$$13y(x) < (1 - 3\varepsilon)12x \quad (22)$$

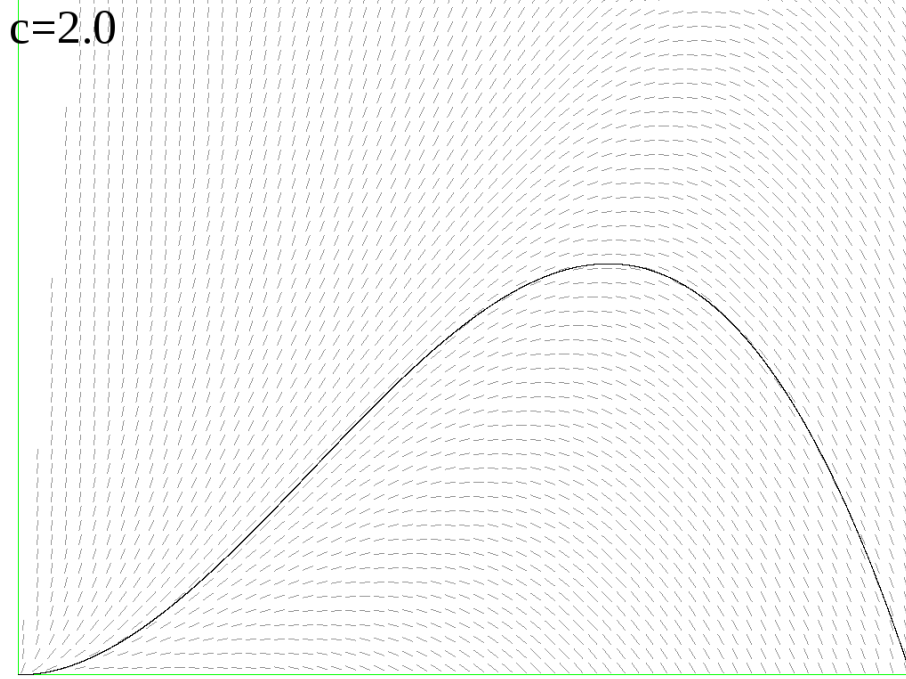


FIGURE 1. Direction field and solution for IVP (21)

holds. Then there exists a  $\tau > 0$  and a strictly decreasing smooth function  $x: [0, \tau] \rightarrow \mathbb{R}$  with  $x(0) = 1$  and  $x(\tau) = x_0$ , such that whp for all  $t$  with  $t/n < \tau$ :

$$X(t) = n x(t/n) + o(n) \tag{23a}$$

$$Y_2(t) = n y(x(t/n)) + o(n) \tag{23b}$$

$$Y_3(t) = n c x(t/n)^3 + o(n). \tag{23c}$$

Moreover, we have the relationship

$$\frac{dx}{dt} = -1 - \frac{y(x)}{x - \frac{13}{12}y(x)} = -\frac{12x - y(x)}{12x - 13y(x)} \tag{23d}$$

*Proof.* For the proof we use Wormald's well-known theorem, which requires some set up and computations. Using the notation of Theorem 5.1 in [40], let

$$\begin{aligned} D &:= \{(t, x, y_2, y_3) \in ]-\varepsilon, c + \varepsilon[^4 \mid (nx, ny_2, ny_3) \text{ is } 2\varepsilon\text{-good}\} \\ C_0 &:= 10 \\ \beta &:= \log^2 n \\ \gamma &:= 3n^{-2} \\ \lambda_1 &:= \frac{\log^5 n}{n} \\ \lambda &:= \frac{\log^{7/3} n}{n^{1/3}}, \end{aligned}$$

Note that  $\lambda > \lambda_1 + C_0 n \gamma$ , and  $\lambda = o(1)$ , as required in Theorem 5.1 in [40].

Obviously, we have  $0 \leq X, Y_2, Y_3 < C_0 n$ .

- (i) Equations 18c, 20b, and 19b, respectively, show that, if  $(t, X(t)/n, Y_2(t)/n, Y_3(t)/n) \in D$ , then, conditioned on  $\mathcal{H}(t)$ , the probability that  $X(t+1) - X(t) \leq \beta$ ,  $Y_2(t+1) - Y_2(t) \leq \beta$ , and  $Y_3(t+1) - Y_3(t) \leq \beta$  hold, is at least  $1 - \gamma$ .
- (ii) The first parts of Lemmas 21, 23, and 22, respectively, show that, if  $(t, x, y_2, y_3) := (t, X(t)/n, Y_2(t)/n, Y_3(t)/n) \in D$ ,

$$\begin{aligned} \left| f(t, x, y_2, y_3) - \mathbf{E}(X(t+1) - X(t) \mid \mathcal{H}(t)) \right| &\leq \lambda_1 \\ \left| g_2(t, x, y_2, y_3) - \mathbf{E}(Y_2(t+1) - Y_2(t) \mid \mathcal{H}(t)) \right| &\leq \lambda_1 \\ \left| g_3(t, x, y_2, y_3) - \mathbf{E}(Y_3(t+1) - Y_3(t) \mid \mathcal{H}(t)) \right| &\leq \lambda_1, \end{aligned}$$

where

$$\begin{aligned} f(t, x, y_2, y_3) &:= -1 - \frac{12y_2}{12x - 13y_2} \\ g_2(t, x, y_2, y_3) &:= \frac{3y_3(t)}{2x(t)} + (1 - f(t, x, y_2, y_3)) \frac{13y_3(t)}{8x(t)} + f(t, x, y_2, y_3) \frac{2y_2(t)}{x(t)} \\ g_3(t, x, y_2, y_3) &:= f(t, x, y_2, y_3) \frac{3y_3(t)}{x(t)}. \end{aligned}$$

- (iii) There exists an  $L$  depending on  $\varepsilon$  such that  $f, g_2, g_3$  are  $L$ -lipschitz continuous on  $D$ .



Let  $x, y_2, y_3$  be the solution to the initial value problem

$$\frac{dx}{dt} = f(t, x(t), y_2(t), y_3(t)) \quad (24a)$$

$$\frac{dy_2}{dt} = g_2(t, x(t), y_2(t), y_3(t)) \quad (24b)$$

$$\frac{dy_3}{dt} = g_3(t, x(t), y_2(t), y_3(t)) \quad (24c)$$

$$x(0) = 1 \quad y_2(0) = 0 \quad y_3(0) = c. \quad (24d)$$

From Wormald's theorem, we conclude that with probability

$$1 - O\left(n\gamma\frac{\beta}{\lambda}e^{-n(\lambda/\beta)^3}\right) = 1 - O\left(\frac{1}{n}\right),$$

it is true that, for all  $t = 0, \dots, \sigma n$ , we have  $X(t) = nx(t/n) + O(\lambda n)$ ,  $Y_2(t) = ny_2(t/n) + O(\lambda n)$ , and  $Y_3(t) = ny_3(t/n) + O(\lambda/n)$ , where  $\sigma = \sigma(n)$  is the supremum over all  $s$  for which the solution to (24) can be extended before reaching within a distance of  $C\lambda$  from the boundary of  $D$ , for a large constant  $C$ .

We now need to study the initial value problem (24). Let us start with the first equation (24a), which we write as

$$\frac{dx}{dt} = -\frac{12x - y_2}{12x - 13y_2},$$

which amounts to

$$-dt = \frac{12x - 13y_2}{12x - y_2} dx = \left(1 - \frac{12y_2}{12x - y_2}\right) dx, \quad (25)$$

The third inequality

$$\frac{dy_3}{dt} = \frac{dx}{dt} \frac{3y_3}{x},$$

is equivalent to

$$\frac{dy_3}{dx} = \frac{3y_3}{x},$$

which immediately integrates to<sup>3</sup>

$$y_3 = cx^3,$$

where the constant before the  $x^3$  is derived from the initial value conditions  $y_3(0) = c$  and  $x(0) = 1$ . Finally, we write the second equation as

$$\frac{dy_2}{dt} = -\frac{y_3}{8x} - \frac{dx}{dt} \frac{13y_3}{8x} + \frac{dx}{dt} \frac{2y_2}{x} = -\frac{y_3}{8x} - \frac{13}{8}cx^2 \frac{dx}{dt} + \frac{2y_2}{x} \frac{dx}{dt}$$

from which we obtain

$$\frac{dy_2}{dx} = -\frac{c}{8}x^2 \frac{dt}{dx} - \frac{13}{8}cx^2 + \frac{2y_2}{x},$$

which, by (25), yields

$$\frac{dy_2}{dx} = \frac{c}{8}x^2 \frac{12x - 13y_2}{12x - y_2} - \frac{13}{8}cx^2 + \frac{2y_2}{x} = \frac{-18cx^4 + 2y_2(12x - y_2)}{x(12x - y_2)}.$$

---

<sup>3</sup>It should be noted that this is the same relationship between  $x$  and  $y_3$  as in the case of classical 3-SAT (see [2]).

which is an ODE of the function  $y_2$  in the variable  $x$ . In fact, with  $y_2(1) = 0$ , we recognize the IVP (21), and thus  $y = y_2$  in the interval on which both are defined.

To summarize, we have  $y_3 = cx^3$ , and  $y_2 = y$  as a function of  $x$  is a solution to the IVP (21), and  $x$  as a function of  $t$  solves the ODE (24a) with boundary condition  $x(0) = 1$ .

From Lemma 25, we know that the solution  $y$  to (21) can be extended to a solution of the IVP define on the full interval  $]0, 1]$ . Moreover,  $\frac{dx}{dt} < 0$  whenever  $13y(x) < 12x$ , so the derivative of  $x$  is strictly negative provided that  $x \geq x_0$ . This implies that the solutions  $x$ ,  $y_2$ ,  $y_3$  to (24) can be extended to the interval  $[0, \tau]$ , where  $\tau$  is the unique number satisfying  $x(\tau) = x_0$ ; in particular we have  $\sigma < \tau$ .

This completes the proof of the lemma.  $\square$

We are now ready to prove Theorem 2.

*Proof of Theorem 2.* Lemma 26 gives the behavior of the parameters  $X(t)$ ,  $Y_2(t)$ , and  $Y_3(t)$  up to an error with high probability for all  $t = 0, \dots, \tau n$ . We need to check that

- (a) the algorithm terminates before  $t$  grows beyond  $\tau n$ ,
- (b) in this region of  $t$ , whp, the algorithm does not produce an empty clause.

For (a), we solve the IVP (21) numerically for  $c = 2.3$ . The solution is drawn in Fig. 2. The figure also shows the line  $12y = 13x$ . For this value of  $c$ , we see that there is an  $\varepsilon > 0$  such that the solution  $y(x)$  to the IVP (21) satisfies  $12y(x) < 13(1 + 2\varepsilon)x$  for all  $x > 2\varepsilon$ ; w.l.o.g., we may assume that  $\varepsilon < 1/9$ . Consequently, the  $x_0$  from Lemma 26 equals  $3\varepsilon$ . Algorithm 1 terminates as soon as  $Y_2(t) + Y_3(t) \leq c'X(t)$ . Thus, by Lemma 26, we have an  $s > \tau$  such that  $x(s) = 1/3 > x_0$ , and that, if we let  $c' := \frac{35}{24}$ , whp, for this  $t := \lceil sn \rceil$

$$\begin{aligned} Y_2(t) + Y_3(t) &= ny(1/3) + nc(1/3)^3 + o(n) \\ &\leq n((1 - \varepsilon)\frac{13}{36} + \frac{1}{9}) + o(n) \leq \frac{17}{36}n = \frac{17}{12} \cdot \frac{1}{3}n \leq c'X(t) - o(n), \end{aligned}$$

if  $n$  is large enough. Thus, the algorithm terminates before the parameters  $X(\cdot)$ ,  $Y_2(\cdot)$ ,  $Y_3(\cdot)$  fail to be  $2\varepsilon$ -good.

It follows that Lemma 24 gives a failure probability of  $o(1/n)$  per iteration, so that the total failure probability is  $o(1)$ . This proves (b) and completes the proof of Theorem 2.  $\square$

## 8. CONCLUSION AND OUTLOOK

We have given an algorithm for  $k$ -iSAT, for  $k = 3$ , which succeeds with high probability on instances for which  $m/n \leq 2.3$ . Though it is conceptually easy to extend to general  $k$  the algorithm and the analysis up to the point where the initial value problem has to be solved, the need to numerically solve a system of  $k - 2$  ordinary differential equations makes it seem improbable that a general formula for the maximal ratio can be derived. Solving the system for small values of  $k$ , we obtain the results shown in Table 2 (we always rounded down generously).

$k$	3	4	5	6	7	8
max. $m/n$	2.3	3.75	6.25	10.5	18.5	32.5

TABLE 2. Performance for different values of  $k$

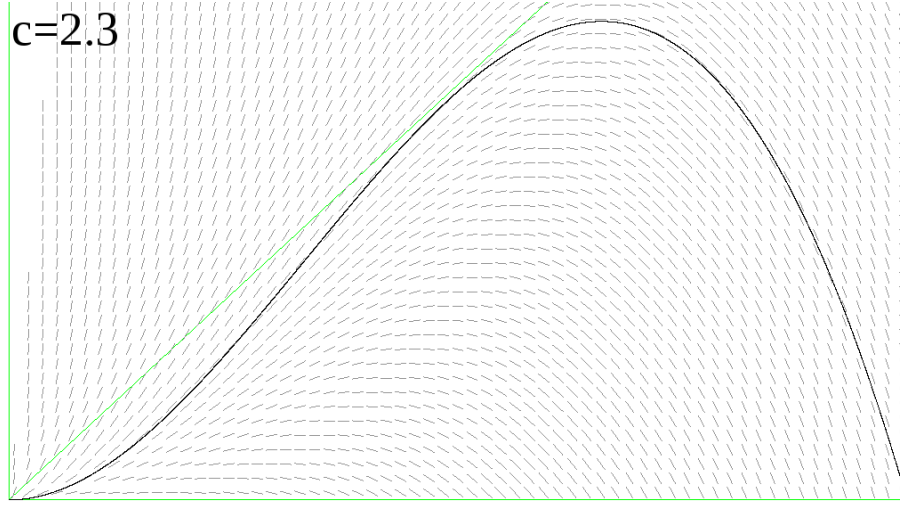


FIGURE 2. Solution of IVP (21) with bounding curves

A couple of issues remain. Firstly, a bound for the ratio above which random 3-iSAT formulas are wpp/whp not satisfiable might be interesting.

Secondly, there might be a threshold behavior for random 2-iSAT, similar to the situation for classical 2-SAT [20, 27]. In fact, we conjecture that there is a threshold at  $c = 3/2$  (the value from Proposition 11). For this it remains to prove that for  $c > 3/2$ , a random 2-iSAT formula with  $m/n = c$  is whp not satisfiable.

Finally, we would like point out that several papers have raised the question of a threshold for random regular 3-iSAT [10, 12, 15, 37]. The situation there is likely to be easier than for iSAT.

We would like to close by thanking various unnamed people for their silent encouragement. The second author expresses his gratitude for the superb working environment at the *Casa del Gelato*.

## REFERENCES

1. Dimitris Achlioptas, *Setting 2 variables at a time yields a new lower bound for random 3-SAT (extended abstract)*, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing (New York), ACM, 2000, pp. 28–37 (electronic). MR 2114514 [3](#), [21](#)
2. ———, *Lower bounds for random 3-SAT via differential equations*, Theoret. Comput. Sci. **265** (2001), no. 1–2, 159–185, Phase transitions in combinatorial problems (Trieste, 1999). MR 1848217 (2002h:68077) [2](#), [3](#), [21](#), [27](#)
3. Dimitris Achlioptas and Yuval Peres, *The threshold for random  $k$ -SAT is  $2^k \log 2 - O(k)$* , J. Amer. Math. Soc. **17** (2004), no. 4, 947–973 (electronic). MR 2083472 (2005g:68137) [2](#)
4. Dimitris Achlioptas and Gregory B. Sorkin, *Optimal myopic algorithms for random 3-SAT*, 41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000), IEEE Comput. Soc. Press, Los Alamitos, CA, 2000, pp. 590–600. MR 1931856 [3](#), [21](#)
5. C. Ansótegui and F. Manyà, *New logical and complexity results for signed-sat*, Proceedings of the 33rd International Symposium on Multiple-Valued Logic, 2003, pp. 181–187. [1](#)

6. Carlos Ansotegui, Ramon Béjar, Alba Cabiscol, and Felip Manyà, *The interface between  $p$  and  $np$  in signed cnf formulas*, Proceedings. 34th International Symposium on Multiple-Valued Logic, 2004, pp. 251–256. [1](#)
7. B. Aspvall, M. Plass, and R. Tarjan, *A linear-time algorithm for testing the truth of certain quantified boolean formulas*, Inf. Process. Lett. **8** (1979), no. 3, 121–123. [2](#), [10](#)
8. Kathrin Ballerstein, *Modeling biological systems and satisfiability of cnf formulas*, Ph.D. thesis, Eidgenössische Technische Hochschule Zürich (ETHZ), in preparation. [2](#)
9. Bernhard Beckert, Reiner Hähnle, and Felip Manyà, *The 2-SAT problem of regular signed CNF formulas*, 30th IEEE International Symposium on Multiple-Valued Logic (ISMVL 2000) (Portland, OR), IEEE Computer Soc., Los Alamitos, CA, 2000, pp. 331–336. MR MR1896571 [1](#)
10. Bernhard Beckert, Reiner Hähnle, and Felip Manyà, *The SAT problem of signed CNF formulas*, Labelled Deduction (M. D’Agostino, D. Basin, D. Gabbay, S. Matthews, and L. Vigano, eds.), Kluwer, Dordrecht, 2000, pp. 59–80. [1](#), [2](#), [29](#)
11. Ramon Béjar, Alba Cabiscol, Cèsar Fernández, Felip Manà, and Carla P. Gomes, *Capturing structure with satisfiability*, Proceedings of CP-2001, LNCS, vol. 2239, Springer, 2001, pp. 137–152. [1](#)
12. Ramon Béjar and Felip Manyà, *Phase transitions in the regular random 3-sat problem.*, ISMIS’99, 1999, pp. M292–300. [2](#), [29](#)
13. ———, *Solving combinatorial problems with regular local search algorithms*, Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR), LNCS, vol. 1705, Springer, 1999, pp. 33–43. [1](#)
14. ———, *Solving the round robin problem using propositional logic*, Proceedings of AAAI-2000, 2000, pp. 262–266. [1](#)
15. Ramón Béjar, Felip Manyà, Alba Cabiscol, Cèsar Fernández, and Carla Gomes, *Regular-SAT: a many-valued approach to solving combinatorial problems*, Discrete Appl. Math. **155** (2007), no. 12, 1613–1626. MR 2341632 (2008f:68135) [1](#), [2](#), [29](#)
16. Katarina Britz, *Interval satisfiability*, Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries (Republic of South Africa), SAICSIT ’04, South African Institute for Computer Scientists and Information Technologists, 2004, pp. 244–248. [1](#)
17. Ming-Te Chao and John Franco, *Probabilistic analysis of two heuristics for the 3-satisfiability problem*, SIAM J. Comput. **15** (1986), no. 4, 1106–1118. MR 861375 (88b:68079) [2](#), [3](#)
18. ———, *Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the  $k$ -satisfiability problem*, Inform. Sci. **51** (1990), no. 3, 289–314. MR 1072035 (91g:68076) [3](#)
19. Victor Chepoi, Nadia Creignou, Miki Hermann, and Gernot Salzer, *The Helly property and satisfiability of Boolean formulas defined on set families*, European J. Combin. **31** (2010), no. 2, 502–516. MR 2565343 (2011c:68070) [1](#), [2](#), [5](#), [10](#), [11](#)
20. V. Chvátal and B. Reed, *Mick gets some (the odds are on his side)*, FOCS, 1992, pp. 620–627. [10](#), [11](#), [29](#)
21. Amin Coja-Oghlan, *A better algorithm for random  $k$ -SAT*, SIAM J. Comput. **39** (2010), no. 7, 2823–2864. MR 2645890 [3](#)
22. J. Downward, *The ins and outs of signalling*, Nature **411** (2001), no. 6839, 759–762. [2](#)
23. Gonzalo Escalada-Imaz and Felip Manyà, *The satisfiability problem for multiple-valued horn formulae*, Proceedings of the IEEE International Symposium on Multiple-Valued Logics, 1994, pp. 250–256. [1](#)
24. Ehud Friedgut, *Sharp thresholds for graph properties, and the  $k$ -sat problem*, Journal of the American Mathematical Society **12** (1999), no. 4, 1017–1054. [3](#)
25. Alan Frieze and Stephen Suen, *Analysis of two simple heuristics on a random instance of  $k$ -SAT*, J. Algorithms **20** (1996), no. 2, 312–355. MR 1379227 (97c:68062) [3](#), [7](#)
26. A.M. Frisch and T.J. Peugniez, *Solving bon-boolean satisfiability problems with stochastic local search*, Proceedings of IJCAI-2001, 2001, pp. 282–288. [1](#)
27. Andreas Goerdt, *A threshold for unsatisfiability*, J. Comput. System Sci. **53** (1996), no. 3, 469–486, 1994 ACM Symposium on Parallel Algorithms and Architectures (Cape May, NJ, 1994). MR MR1423858 (98i:03012) [29](#)
28. Geoffrey R. Grimmett and David R. Stirzaker, *Probability and random processes*, third ed., Oxford University Press, New York, 2001. MR 2059709 (2004m:60002) [32](#)

29. Reiner Hähnle, *Towards an efficient tableau proof procedure for multiple-valued logics*, Lecture Notes in Computer Science **533/1991** (1991), 248–260, DOI: 10.1007/3-540-54487-9\_62. [1](#)
30. Utz-Uwe Haus, Kathrin Niermann, Klaus Truemper, and Robert Weismantel, *Logic integer programming models for signaling networks*, Journal of Computational Biology **16** (2009), no. 5, 725–743. [2](#)
31. Svante Janson, Tomasz Łuczak, and Andrzej Ruciński, *Random graphs*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley-Interscience, New York, 2000. MR 1782847 (2001k:05180) [16](#)
32. Joyce Justicz, Edward R. Scheinerman, and Peter M. Winkler, *Random intervals*, Amer. Math. Monthly **97** (1990), no. 10, 881–889. MR MR1079974 (91m:60023) [1](#), [9](#)
33. Stuart A. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets*, J. Theor. Biol. **22** (1969), no. 3, 437–467. [2](#)
34. Steffen Klamt, Julio Saez-Rodriguez, Jonathan A. Lindquist, Luca Simeoni, and Ernst Dieter Gilles, *A methodology for the structural and functional analysis of signaling and regulatory networks.*, BMC Bioinformatics **7** (2006), no. 56, available from <http://www.biomedcentral.com/1471-2105/7/56>. [2](#)
35. Jan Łukasiewicz, *O logice trjwartociowej*, Ruch Filozoficzny (1920), 169–171, In polish. [1](#)
36. Felip Manyà, *The 2-SAT problem in signed CNF formulas*, Multiple-Valued Logic **5** (2000), no. 4, 307–325. [1](#)
37. Felip Manyà, Ramon Béjar, and Gonzalo Escalada-Imaz, *The satisfiability problem in regular cnf-formulas*, Soft Computing: A Fusion of Foundations, Methodologies and Applications **2** (1998), no. 3, 116–123. [2](#), [29](#)
38. E. R. Scheinerman, *Random interval graphs*, Combinatorica **8** (1988), no. 4, 357–371. MR MR981893 (90a:05171) [1](#), [4](#), [7](#), [9](#), [11](#)
39. Nicholas C. Wormald, *Differential equations for random processes and random graphs*, Ann. Appl. Probab. **5** (1995), no. 4, 1217–1235. MR MR1384372 (97c:05139) [3](#)
40. ———, *The differential equation method for random graph processes and greedy algorithms*, Lectures on Approximation and Randomized Algorithms (M. Karonski and H. J. Proemel, eds.), PWN, Warsaw, 1999, pp. 73–155. [3](#), [26](#)

## APPENDIX A. DEFERRED PROOFS

**A.1. Computations for Lemma 6.** For (a), we compute

$$\begin{aligned}
 1 - \mathbf{E} P &= \mathbf{E}(2X(1 - X)) = \int 2t(1 - t) dF(t) \\
 &= \int_{[0, 1/2[} 2t(1 - t) \partial_t F(t) dt + 2t(1 - t) \Big|_{t=1/2} \cdot \frac{1}{2} + \int_{]1/2, 1]} 2t(1 - t) \partial_t F(t) dt \\
 &= \int_0^{1/2} 2t(1 - t) 2t dt + \frac{1}{4} + \int_{1/2}^1 2t(1 - t) 2(1 - t) dt \\
 &= \frac{5}{48} + \frac{1}{4} + \frac{5}{48} = \frac{11}{24}.
 \end{aligned}$$

For (b), we compute

$$\begin{aligned}
\mathbf{E}(X^2(1-X)^2) &= \int t^2(1-t)^2 dF(t) \\
&= t^2(1-t)^2 \Big|_{t=1/2} \cdot \frac{1}{2} + \int_{[0,1/2[} t^2(1-t)^2 \partial_t F(t) dt + \int_{]1/2,1]} t^2(1-t)^2 \partial_t F(t) dt \\
&= \frac{1}{2^5} + \int_{[0,1/2[} t^2(1-t)^2 2t dt + \int_{]1/2,1]} t^2(1-t)^2 2(1-t) dt \\
&= \frac{1}{2^5} + 4 \int_0^{1/2} t^3(1-t)^2 dt \\
&= \frac{1}{2^5} + 4 \left( \frac{1}{4} t^4(1-t)^2 \Big|_{t=0}^{1/2} + \frac{1}{10} t^5(1-t) \Big|_{t=0}^{1/2} + \frac{1}{60} t^6 \Big|_{t=0}^{1/2} \right) \\
&= \frac{1}{2^5} + 4 \left( \frac{1}{4} \frac{1}{2^6} + \frac{1}{10} \frac{1}{2^6} + \frac{1}{60} \frac{1}{2^6} \right) = \frac{1}{2^5} + \frac{1}{2^6} \left( 1 + \frac{2}{5} + \frac{1}{15} \right) = \frac{1}{2^5} + \frac{22}{15 \cdot 2^6} = \frac{1}{2^5} + \frac{11}{15 \cdot 2^5} = \frac{15+11}{15 \cdot 2^5} = \frac{13}{15 \cdot 2^4}.
\end{aligned}$$

Hence, using (a), we obtain

$$\mathbf{E}(P^2) = 1 - 2(1 - \mathbf{E} P) + 4 \mathbf{E} X^2(1-X)^2 = 1 - \frac{11}{12} + \frac{13}{60} = \frac{18}{60} = \frac{3}{10}.$$

**A.2. Proof of Lemma 12.** The proof is taken almost word for word from Grimmett & Stirzaker [28], Theorem 11.3.17, with some changes due to the discrete arrival- and servicing points.

We say that the *sons* of a customer Paul are those customers arriving in the time interval in which Paul is serviced. Paul's *family* consists of himself and all of his descendants.

Fix a time interval  $j$  in which the queue is not empty and denote by  $X$  the size of the family of the customer served at that time interval. We have the relation

$$X = 1 + \sum_{i=1}^{B(j+1)} X_i,$$

where  $X_i$  denotes the family size of the  $i$ 'th customer arriving in the time interval  $j$ .

The important observation now is that the family sizes are iid because the  $B(j)$  are iid, and that the  $X_i$  are independent of  $B(j+1)$ . Consequently, for the common probability generating function  $y$  of  $X$  and the  $X_i$ , we have

$$y(x) = x g_B(y(x)). \quad (*)$$

The length first busy period coincides with sum of the family sizes of the  $a$  customers arriving in the first time interval. Thus, we obtain

$$h(x) = y(x)^a. \quad (**)$$

Solving  $(*)$  for  $x$  and inserting into  $(**)$ , we obtain

$$h\left(\frac{g_B(y(x))}{y(x)}\right) = y(x)^a. \quad (***)$$

If  $y(0) = 0$ , then  $B = 0$ , and thus  $h(y) = y^a$ , which coincides with equation (5a). Otherwise, by  $(***)$ , equation (5a) holds for all  $y$  in the interval  $[y(0), y(1)]$ , and thus for all  $y$  for which the power series on both sides of the equality sign converge.

We derive the statement about the mean length of the first busy period by differentiating (5a), and possibly invoking Abel's Theorem to evaluate the power series at the point 1.



Finally, the statement about the tail probability follows directly from the standard exponential moment argument: If  $y \geq g_B(y) > 0$ , then, with  $x := y/g_B(y) \geq 1$ , we have

$$\mathbf{P}[Z \geq \alpha] = \mathbf{P}[x^Z \geq x^\alpha] \leq \frac{\mathbf{E} x^Z}{x^\alpha} = \frac{h(x)}{x^\alpha} = \frac{y^a}{(y/g_B(y))^\alpha} = \frac{g_B(y)^\alpha}{y^{\alpha-a}},$$

as claimed.

**A.3. Computations for Lemma 14.** Computations regarding equation (\*\*):

$$\begin{aligned} \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 u - (\alpha - a) \frac{1}{u+1} &= 0 \\ \alpha r(u+1) + \frac{12^3}{13^{2.5}} \alpha r^2 u(u+1) - (\alpha - a) &= 0 \\ \left( \frac{12^3}{13^{2.5}} \alpha r^2 \right) u^2 + \left( \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 \right) u - ((1-r)\alpha - a) &= 0 \\ u = - \frac{\left( \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 \right) \pm \sqrt{\left( \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 \right)^2 + 4((1-r)\alpha - a) \left( \frac{12^3}{13^{2.5}} \alpha r^2 \right)}}{2 \cdot \left( \frac{12^3}{13^{2.5}} \alpha r^2 \right)} \end{aligned}$$

We need to be close to 0, so we take the “ $\pm$ ” = “ $-$ ”:

$$\begin{aligned} u_r &:= \frac{-\left( \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 \right) + \sqrt{\left( \alpha r + \frac{12^3}{13^{2.5}} \alpha r^2 \right)^2 + 4((1-r)\alpha - a) \frac{12^3}{13^{2.5}} \alpha r^2}}{2 \cdot \frac{12^3}{13^{2.5}} \alpha r^2} \\ &= \frac{-\left( 1 + \frac{12^3}{13^{2.5}} r \right) + \sqrt{\left( 1 + \frac{12^3}{13^{2.5}} r \right)^2 + 4(1-r-a/\alpha) \frac{12^3}{13^{2.5}}}}{2 \cdot \frac{12^3}{13^{2.5}} r} \\ &= \frac{-\left( 1 + \frac{12^3}{13^{2.5}} r \right) + \sqrt{\left( 1 + \frac{12^3}{13^{2.5}} r \right)^2 - 4r \frac{12^3}{13^{2.5}} + 4(1-a/\alpha) \frac{12^3}{13^{2.5}}}}{2 \cdot \frac{12^3}{13^{2.5}} r} \\ &= \frac{-\left( 1 + \frac{12^3}{13^{2.5}} r \right) + \sqrt{\left( 1 - \frac{12^3}{13^{2.5}} r \right)^2 + 4(1-a/\alpha) \frac{12^3}{13^{2.5}}}}{2 \cdot \frac{12^3}{13^{2.5}} r} \\ &= \frac{-\left( 1 + \frac{12^3}{5 \cdot 13^2} r \right) + \sqrt{\left( 1 - \frac{12^3}{5 \cdot 13^2} r \right)^2 + \frac{4 \cdot 12^3}{5 \cdot 13^2} - \frac{4 \cdot 12^3}{5 \cdot 13^2} \cdot \frac{a}{\alpha}}}{\frac{2 \cdot 12^3}{5 \cdot 13^2} r} \\ &= \frac{-\left( 1 + \frac{12^3}{5 \cdot 13^2} r \right) + \sqrt{\left( 1 - \frac{12^3}{5 \cdot 13^2} r \right)^2 + \frac{4 \cdot 12^3}{5 \cdot 13^2}}}{\frac{2 \cdot 12^3}{5 \cdot 13^2} r} - O(a/\alpha), \end{aligned}$$

with an absolute constant in the  $O(\cdot)$ , because  $a \leq \alpha$  and  $1/2 \leq r \leq 1$ .

Computation regarding equation (\*\*\*\*):

$$\begin{aligned} \frac{(*)}{\alpha}(u_r) &= \frac{\alpha r u + \frac{12^2 \cdot 3 \cdot 2}{13^2 \cdot 5} \alpha r^2 u^2 - (\alpha - a) \log(u + 1)}{\alpha} \Big|_{u:=u_r} \\ &= r u_r + \frac{6 \cdot 12^2}{5 \cdot 13^2} r^2 u_r^2 - (1 - a/\alpha) \log(u_r + 1) \\ &= r u_r + \frac{6 \cdot 12^2}{5 \cdot 13^2} r^2 u_r^2 - \log(u_r + 1) + O(a/\alpha), \end{aligned}$$

with an absolute constant in the  $O(\cdot)$ , because  $u_r + 1 \leq 2$ .

**A.4. Proof of Lemma 17.** Suppose that the  $B(j)$  are represented as a sum as in (10) above, and define

$$B^\pm(j) := \sum_{i=1}^{m^\pm} \mathbf{I}\left[U(j, i) \leq \frac{P(j)}{n - (\pm z)}\right].$$

Then the  $B^+(j)$ ,  $j = 1, 2, 3, \dots$ , are iid, so that Lemma 16 is applicable. The same is true for the  $B^-(j)$ ,  $j = 1, 2, 3, \dots$ . We clearly have, with probability  $1 - O(n^{-r})$ ,

$$B^-(j) \leq B(j) \leq B^+(j) \quad \text{for all } j = 1, \dots, z.$$

Defining two queues  $Q^\pm(j)$  based on the  $B^\pm(j)$  and respective lengths of first busy periods  $Z^\pm$ , we obtain, with probability  $1 - O(n^{-r})$

$$Z^- \leq Z \leq Z^+, \tag{*}$$

where we have also used that  $Z^\pm \leq z$  with probability  $1 - O(n^{-r})$  (Lemma 16).

Denote by  $E$  the event that  $(*)$  holds. If  $(*)$  does not hold, we still have  $Z = O(n)$  by (11), so that we obtain

$$\mathbf{E} Z = \mathbf{E}(Z | E) \mathbf{P}(E) + \mathbf{E}(Z | \bar{E}) \mathbf{P}(\bar{E}) \leq \mathbf{E}(Z^+ | E) \mathbf{P}(E) + O(n^{1-r}) \leq \mathbf{E}(Z^+) + O(n^{1-r}).$$

For the lower bound, we similarly have

$$\mathbf{E} Z \geq \mathbf{E}(\mathbf{I}(E) Z^-) = \mathbf{E}(Z^-) - \mathbf{E}(\mathbf{I}(\bar{E}) Z^-)$$

Clearly,  $\mathbf{E}(\mathbf{I}(\bar{E}) Z^-) \leq z \mathbf{P}(\bar{E}) + \mathbf{E}(\mathbf{I}(\bar{E}) Z^- \mathbf{I}[Z^- > z]) = z \mathbf{P}(\bar{E}) + m O(n^{-r}) = O(n^{1-r})$

Thus we conclude that  $\mathbf{E} Z \geq \mathbf{E} Z^- - O(n^{1-r})$ .

For the tail estimate, we use  $Z^+$ :

$$\begin{aligned} \mathbf{P}[Z \geq \alpha] &\leq \mathbf{P}[Z \geq \alpha \ \& \ Z \leq Z^+] + \mathbf{P}[Z \geq \alpha \ \& \ Z > Z^+] \\ &\leq \mathbf{P}[Z^+ \geq \alpha] + \mathbf{P}[Z > Z^+] \leq e^{-\delta \alpha} + O(n^{-r}) \end{aligned}$$

by Lemma 16.

**A.5. Computations for Lemma 25.**

For the proof of Claim (b). Let  $g(x, y) := \frac{-18cx^4 + 2y(12x - y)}{x(12x - y)}$ , the right hand side of the ODE (21a). As mentioned in the proof of the lemma, we show  $g(x, y(x)) > 6 = \frac{dy}{dx}$ , for  $0 < x \leq 4/5$ . We compute

$$\begin{aligned} g(x, y(x)) &= \frac{-18cx^4 + 2 \cdot 6x(12x - 6x)}{x(12x - 6x)} = \frac{-18cx^2 + 2 \cdot 6(12 - 6)}{(12 - 6)} = \frac{-18cx^2 + 72}{6} \\ &= -3cx^2 + 12 \underset{c \leq 3}{\geq} -9x^2 + 12 \geq -9(4/5)^2 + 12 = 12 - \frac{9 \cdot 16}{25} = \frac{25 \cdot 12 - 9 \cdot 16}{25} \\ &= \frac{12(25 - 3 \cdot 4)}{25} = \frac{12 \cdot 13}{25} > 6. \end{aligned}$$

For the proof of Claim (a). Let  $g(x, y)$  as above. As mentioned in the proof of the lemma, we show  $g(x, y(x)) > -5 = \frac{dy}{dx}$ , for  $4/5 \leq x \leq 1$ . To show that

$$g(x, y(x)) = \frac{-18cx^4 + 2 \cdot 5(1 - x)(12x - 5(1 - x))}{x(12x - 5(1 - x))} > -5,$$

we compute

$$\begin{aligned} &-18cx^4 + 2 \cdot 5(1 - x)(12x - 5(1 - x)) + 5x(12x - 5(1 - x)) \\ &= -18cx^4 + 10(1 - x)(17x - 5) + 5x(17x - 5) = -18cx^4 + (10 - 5x)(17x - 5) \\ &= -18cx^4 - 85x^2 + 195x - 50 \underset{c \leq 3}{\geq} -54x^4 - 85x^2 + 195x - 50. \end{aligned}$$

The derivative  $-216x^3 - 170x + 195$  of the last polynomial is strictly decreasing, and evaluating it at  $4/5$  gives  $-216(4/5)^3 - 170 \cdot 4/5 + 195 \approx -51.592 < 0$ . Thus, it suffices to check the inequality  $-54x^4 - 85x^2 + 195x - 50 > 0$  for  $x = 1$ :  $-54 - 85 + 195 - 50 = 6 > 0$ .

INSTITUTE FOR OPERATIONS RESEARCH, DEPARTMENT OF MATHEMATICS, ETH ZURICH, RÄMISTRASSE 101, 8092 ZURICH, SWITZERLAND

*E-mail address:* kathrin.ballerstein@ifor.math.ethz.ch

FAKULTÄT FÜR MATHEMATIK, OTTO-VON-GUERICKE-UNIVERSITÄT MAGDEBURG, UNIVERSITÄTSPLATZ 2, 39106 MAGDEBURG, GERMANY

*E-mail address:* dirk.theis@ovgu.de